

Universidad Técnica Nacional Sede  
Guanacaste, Liberia

Ingeniería en Tecnologías de Información

UTN en Guanacaste

Documentación para el usuario final

Estudiantes:

Christian Paniagua Castro 504590528

Steven Baltodano Ugarte 504640801

Reyman Barquero Ramirez 504640986

Bryan Vega Ordoñez 504480650

Felipe Sandoval Chaverri 504260417

II Cuatrimestre 2025

# Sistema CRM Bastos Cañas

<b>Bitacora .....</b>	3
<b>Estructura del software web CRM .....</b>	4
<b>Script de la bd .....</b>	727
<b>Estructura del Proyecto formato png .....</b>	748
<b>Descripción del proyecto .....</b>	749
<b>Extensiones utilizadas en el desarrollo del sistema .....</b>	751
<b>Acceso y Reglas .....</b>	751
<b>API WhatsApp -&gt; Pasos.....</b>	752
<b>Requerimientos de hardware .....</b>	754
<b>Diagrama de entidad-relación .....</b>	755
<b>Diagrama de clases UML.....</b>	756
<b>Diagrama de secuencia Cliente externo .....</b>	757

## Bitacora

**Tipo de servidor:** Físico

**Hipervisor:** Tipo 1 (bare-metal)

**Plataforma de virtualización:** Citrix XenServer

**Dirección IP del host:** 10.50.80.199

**Puerto físico utilizado:** GigabitEthernet0/1

**Contraseña del servidor:** 202502

**Rol principal:** Servidor web y de base de datos (ejecuta XAMPP).

---

## Configuración de la Máquina Virtual

Dentro de la VM se encuentra instalado **XAMPP**, que incluye:

**Apache** (escuchando en el puerto 80)

**MySQL** (escuchando en el puerto 3306)

**PHP**

**GitKraken** (cliente de Git)

**Visual Studio Code** (IDE)

Apache y MySQL están en ejecución, lo que permite al servidor atender solicitudes web y gestionar la base de datos.

---

## Sockets y Servicios Utilizados

**10.50.80.218:80** → Servicio **Apache**, permite acceder al sistema web desde un navegador dentro de la red local.

**127.0.0.1:3306** → Servicio **MySQL**, gestiona la base de datos y solo acepta conexiones locales dentro de la máquina virtual.

**10.50.80.218:3389** → Servicio **RDP (Escritorio Remoto)**, utilizado para conectarse a la máquina virtual desde otro equipo.

---

## Ubicación del Proyecto

**CRM principal:** C:\xampp\htdocs\CRM\_INT

**Formulario web:** C:\xampp\htdocs\CRM\_FORMULARIO

## Estructura del software web CRM

```
CRM_INT/
└─ CRM/
    └─ bridge-node/
        └─ .env.example
```

```
# Puerto de la API
PORT=3001
# Ruta para guardar sesión (persistirá credenciales)
WHATS_SESSION_PATH=./session
```

```
└─ install-service.cjs
```

```
var Service = require('node-windows').Service;

// Crear un nuevo objeto de servicio
var svc = new Service({
    name: 'WhatsApp Bridge CRM',
    description: 'Servicio de WhatsApp para el CRM de Bastos',
    script: 'C:\\xampp\\htdocs\\CRM_INT\\CRM\\bridge-node\\server.js',
    nodeOptions: [
        '--harmony',
        '--max_old_space_size=4096'
    ],
    env: {
        name: "NODE_ENV",
        value: "production"
    }
});

// Escuchar el evento 'install' que se dispara cuando el servicio se
instala
svc.on('install', function(){
    console.log('☒ Servicio WhatsApp Bridge instalado correctamente');
    console.log('Iniciando servicio...');
    svc.start();
});

// Escuchar el evento 'start' que se dispara cuando el servicio inicia
svc.on('start', function(){
    console.log('☒ Servicio WhatsApp Bridge iniciado');
    console.log('El servicio ahora correrá automáticamente al iniciar
Windows');
    console.log('URL: http://localhost:3001');
});
```

```
// Instalar el servicio
console.log('Instalando servicio WhatsApp Bridge... ');
svc.install();
```

```
|   └── package-lock.json
{
  "name": "whatsapp-local",
  "version": "1.0.0",
  "lockfileVersion": 3,
  "requires": true,
  "packages": {
    "": {
      "name": "whatsapp-local",
      "version": "1.0.0",
      "dependencies": {
        "cors": "^2.8.5",
        "dotenv": "^16.4.5",
        "express": "^4.19.2",
        "node-windows": "^1.0.0-beta.8",
        "qrcode": "^1.5.4",
        "whatsapp-web.js": "^1.26.0"
      }
    },
    "node_modules/@pedroslopez/moduleraid": {
      "version": "5.0.2",
      "resolved": "https://registry.npmjs.org/@pedroslopez/moduleraid/-/moduleraid-5.0.2.tgz",
      "integrity": "sha512-wtnBAETBVYZ9GvcbgdswRVSLkFkYAGv1KzwBBTeRXvGT9sb9cPllOgFFWX Cn9PyARQ0H+ljz6mmoRrGateUDxQ==",
      "license": "MIT"
    },
    "node_modules/@types/node": {
      "version": "24.2.1",
      "resolved": "https://registry.npmjs.org/@types/node/-/node-24.2.1.tgz",
      "integrity": "sha512-DRh5K+ka5eJic8CjH7td8QpYEV6Zo10gfRkjHCO3weqZHWDtAaSTFtl4+VMq OJ4N5jcuhZ9/l+yy8rVgw7BQeQ==",
      "license": "MIT",
      "optional": true,
      "dependencies": {
        "undici-types": "~7.10.0"
```

```

    },
    "node_modules/@types/yauzl": {
      "version": "2.10.3",
      "resolved": "https://registry.npmjs.org/@types/yauzl/-/yauzl-2.10.3.tgz",
      "integrity": "sha512-oJoftv0LSuaDZE3Le4DbKX+KS9G36NzOeSap90UIK0yMA/NhKJhqISGtNDORNRalbQfzjXDrQa0ytJ6mNRGz/Q==",
      "license": "MIT",
      "optional": true,
      "dependencies": {
        "@types/node": "*"
      }
    },
    "node_modules/accepts": {
      "version": "1.3.8",
      "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.8.tgz",
      "integrity": "sha512-PYAthTa2m2VKxuvSD3DPC/Gy+U+sOA1LAuT8mkmRuvw+NACsaeXEQ+NHcVF7rONI6qcaxV3Uuemwawk+7+SJLw==",
      "license": "MIT",
      "dependencies": {
        "mime-types": "~2.1.34",
        "negotiator": "0.6.3"
      },
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/agent-base": {
      "version": "6.0.2",
      "resolved": "https://registry.npmjs.org/agent-base/-/agent-base-6.0.2.tgz",
      "integrity": "sha512-RZNwNclF7+MS/8bDg70amg32dyeZGZxiDuQmZxKLAIQjr3jGyLx+4Kkk58UO7D2QdgFIQCovuSuZESne6RG6XQ==",
      "license": "MIT",
      "dependencies": {
        "debug": "4"
      },
      "engines": {
        "node": ">= 6.0.0"
      }
    }
  }
}

```

```

},
"node_modules/agent-base/node_modules/debug": {
  "version": "4.4.1",
  "resolved": "https://registry.npmjs.org/debug/-/debug-4.4.1.tgz",
  "integrity": "sha512-KcKCqiftBJcZr++7ykoDIEwSa3XWowTfNPo92BYxjXiyYEvrUQh2aLyhxBCwww+heortUFxEJYcRzosstTEBYQ==",
  "license": "MIT",
  "dependencies": {
    "ms": "^2.1.3"
  },
  "engines": {
    "node": ">=6.0"
  },
  "peerDependenciesMeta": {
    "supports-color": {
      "optional": true
    }
  }
},
"node_modules/agent-base/node_modules/ms": {
  "version": "2.1.3",
  "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
  "integrity": "sha512-6FIzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTIA==",
  "license": "MIT"
},
"node_modules/ansi-regex": {
  "version": "5.0.1",
  "resolved": "https://registry.npmjs.org/ansi-regex/-/ansi-regex-5.0.1.tgz",
  "integrity": "sha512-quJQXITSUGL2LH9SUxO8VwsY4soanhgo6LNSm84E1LBcE8s3O0wpdiRzyR9z/ZZJMIMWv37qOOB9pdJIMUEKFQ==",
  "license": "MIT",
  "engines": {
    "node": ">=8"
  }
},
"node_modules/ansi-styles": {
  "version": "4.3.0",
  "resolved": "https://registry.npmjs.org/ansi-styles/-/ansi-styles-4.3.0.tgz",

```

```

  "integrity": "sha512-
zbB9rCJAT1rbjiVDb2hqKFHNYLxgtk8NURxZ3IZwD3F6NtxbXZQCnnSi1Lkx+I
DohdPIFp222wVALIheZJQSEg==",
  "license": "MIT",
  "dependencies": {
    "color-convert": "^2.0.1"
  },
  "engines": {
    "node": ">=8"
  },
  "funding": {
    "url": "https://github.com/chalk/ansi-styles?sponsor=1"
  },
  "node_modules/archiver": {
    "version": "5.3.2",
    "resolved": "https://registry.npmjs.org/archiver/-/archiver-5.3.2.tgz",
    "integrity": "sha512-
+25nxyyznAXF7Nef3y0EbBeqmGZgeN/BxHX29Rs39djAfaFalmQ89SE6CWyD
CHzGL0yt/ycBtNOmGTW0FyGWNw==",
    "license": "MIT",
    "optional": true,
    "dependencies": {
      "archiver-utils": "^2.1.0",
      "async": "^3.2.4",
      "buffer-crc32": "^0.2.1",
      "readable-stream": "^3.6.0",
      "readdir-glob": "^1.1.2",
      "tar-stream": "^2.2.0",
      "zip-stream": "^4.1.0"
    },
    "engines": {
      "node": ">= 10"
    }
  },
  "node_modules/archiver-utils": {
    "version": "2.1.0",
    "resolved": "https://registry.npmjs.org/archiver-utils/-/archiver-utils-2.1.0.tgz",
    "integrity": "sha512-
bEL/yUb/fNNINTuUz979Z0Yg5L+LzLxGJz8x79lYmR54fmTlb6ob/hNQgkQnIU
DWIFjZVQwl9Xs356l6BAMHfw==",
  }
}

```

```

"license": "MIT",
"optional": true,
"dependencies": {
  "glob": "^7.1.4",
  "graceful-fs": "^4.2.0",
  "lazystream": "^1.0.0",
  "lodash.defaults": "^4.2.0",
  "lodash.difference": "^4.5.0",
  "lodash.flatten": "^4.4.0",
  "lodash.isplainobject": "^4.0.6",
  "lodash.union": "^4.6.0",
  "normalize-path": "^3.0.0",
  "readable-stream": "^2.0.0"
},
"engines": {
  "node": ">= 6"
}
},
"node_modules/archiver-utils/node_modules/readable-stream": {
  "version": "2.3.8",
  "resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-2.3.8.tgz",
  "integrity": "sha512-8p0AUk4XODglewSi0l8Epjs+EVnWiK7NoDIEGU0HhE7+ZyY8D1IMY7odu5lRrFXGg71L15KG8QrPmum45RTtdA==",
  "license": "MIT",
  "optional": true,
  "dependencies": {
    "core-util-is": "~1.0.0",
    "inherits": "~2.0.3",
    "isarray": "~1.0.0",
    "process-nextick-args": "~2.0.0",
    "safe-buffer": "~5.1.1",
    "string_decoder": "~1.1.1",
    "util-deprecate": "~1.0.1"
  }
},
"node_modules/archiver-utils/node_modules/safe-buffer": {
  "version": "5.1.2",
  "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.2.tgz",

```

```

    "integrity": "sha512-Gd2UZBJDkXIY7GbJxfsE8/nvKkUEU1G38c1siN6QP6a9PT9MmHB8GnpSCSmMJSoF8LOlrt8ud/wPtojys4G6+g==",
    "license": "MIT",
    "optional": true
},
"node_modules/archiver-utils/node_modules/string_decoder": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-1.1.1.tgz",
    "integrity": "sha512-n/ShnDi6FHbbVfviro+WojiFzv+s8MPMHBCzVePfUpDJLwoLT0ht1I4YwBCbi8pJAveEEdnkHyPyTP/mzRfwg==",
    "license": "MIT",
    "optional": true,
    "dependencies": {
        "safe-buffer": "~5.1.0"
    }
},
"node_modules/array-flatten": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/array-flatten/-/array-flatten-1.1.1.tgz",
    "integrity": "sha512-PCVAQswWemu6UdxsDFFX/+gVeYqKAod3D3UVm91jHwynguOwAvYPhx8nNIM++NqRcK6CxxpUafjmhdKiHibqg==",
    "license": "MIT"
},
"node_modules/async": {
    "version": "3.2.6",
    "resolved": "https://registry.npmjs.org/async/-/async-3.2.6.tgz",
    "integrity": "sha512-htCUDlxxyCLMgaM3xXg0C0LW2xqfuQ6p05pCEIsXuyQ+a1koYKTuBMzRNwmybfLgvJDMd0r1LTn4+E0Ti6C2AA==",
    "license": "MIT",
    "optional": true
},
"node_modules/balanced-match": {
    "version": "1.0.2",
    "resolved": "https://registry.npmjs.org/balanced-match/-/balanced-match-1.0.2.tgz",

```

```

    "integrity": "sha512-
3oSeUO0TMV67hN1AmbXsK4yaqU7tjiHlbxRDZOpH0KW9+CeX4bRAaX0Anxt
0tx2MrpRpWwQaPwlISEJhYU5Pw==",
    "license": "MIT"
},
"node_modules/base64-js": {
    "version": "1.5.1",
    "resolved": "https://registry.npmjs.org/base64-js/-/base64-js-1.5.1.tgz",
    "integrity": "sha512-
AKpaYIHn8t4SVbOHCy+b5+KKgvR4vrsD8vbvrbiQJps7fKDTkjkdry6ji0rUJjC0k
zbNePLwzxq8iypo41qeWA==",
    "funding": [
        {
            "type": "github",
            "url": "https://github.com/sponsors/feross"
        },
        {
            "type": "patreon",
            "url": "https://www.patreon.com/feross"
        },
        {
            "type": "consulting",
            "url": "https://feross.org/support"
        }
    ],
    "license": "MIT"
},
"node_modules/big-integer": {
    "version": "1.6.52",
    "resolved": "https://registry.npmjs.org/big-integer/-/big-integer-1.6.52.tgz",
    "integrity": "sha512-
QxD8cf2eVqJOz63z6JIN9BzvVs/dlySa5HGSBH5xtR8dPteIRQnBxxKqkNTiT
6jbDTF6jAfrd4oMcND9RGbQg==",
    "license": "Unlicense",
    "optional": true,
    "engines": {
        "node": ">=0.6"
    }
},
"node_modules/binary": {
    "version": "0.3.0",
    "resolved": "https://registry.npmjs.org/binary/-/binary-0.3.0.tgz",

```

```

    "integrity": "sha512-D4H1y5KYwpJgK8wk1Cue5LLPgmwHKYSChkbsPQg5JtVuR5uIGckxfR62H3AE9UDkdMC8yyXlqYihuz3Aqg2XZg==",
    "license": "MIT",
    "optional": true,
    "dependencies": {
        "buffers": "~0.1.1",
        "chainsaw": "~0.1.0"
    },
    "engines": {
        "node": "*"
    },
    "node_modules/bl": {
        "version": "4.1.0",
        "resolved": "https://registry.npmjs.org/bl/-/bl-4.1.0.tgz",
        "integrity": "sha512-1W07cM9gS6DcLperZfFSj+bWLtaPGSOHWhPiGzXmvVJbRLdG82sH/Kn8EtW1VqWVA54AKf2h5k5BbnIbwF3h6w==",
        "license": "MIT",
        "dependencies": {
            "buffer": "^5.5.0",
            "inherits": "^2.0.4",
            "readable-stream": "^3.4.0"
        }
    },
    "node_modules/bluebird": {
        "version": "3.4.7",
        "resolved": "https://registry.npmjs.org/bluebird/-/bluebird-3.4.7.tgz",
        "integrity": "sha512-iD3898SR7sWVRHbiQv+sHUtHnMvC1o3nW5rAcqnq3uOn07DSAppZYUkIGsIDz6gXC7HfunPe7YVBgoEJASPcHA==",
        "license": "MIT",
        "optional": true
    },
    "node_modules/body-parser": {
        "version": "1.20.3",
        "resolved": "https://registry.npmjs.org/body-parser/-/body-parser-1.20.3.tgz",
        "integrity": "sha512-7rAxByjUMqQ3/bHJy7D6OGXvx/MMc4IqBn/X0fcM1QUcAltpZrBEYhWGem+tzXH90c+G01ypMcYJB09Y30203g==",
        "license": "MIT",

```

```

"dependencies": {
    "bytes": "3.1.2",
    "content-type": "~1.0.5",
    "debug": "2.6.9",
    "depd": "2.0.0",
    "destroy": "1.2.0",
    "http-errors": "2.0.0",
    "iconv-lite": "0.4.24",
    "on-finished": "2.4.1",
    "qs": "6.13.0",
    "raw-body": "2.5.2",
    "type-is": "~1.6.18",
    "unpipe": "1.0.0"
},
"engines": {
    "node": ">= 0.8",
    "npm": "1.2.8000 || >= 1.4.16"
}
},
"node_modules/brace-expansion": {
    "version": "1.1.12",
    "resolved": "https://registry.npmjs.org/brace-expansion/-/brace-expansion-1.1.12.tgz",
    "integrity": "sha512-9T9UjW3r0UW5c1Q7GTwiptXwhvYmEzFhzMfZ9H7FQWt+uZePjZPjBP/W1ZEyZ1twGWom5/56TF4IPcqjnDHcg==",
    "license": "MIT",
    "dependencies": {
        "balanced-match": "^1.0.0",
        "concat-map": "0.0.1"
    }
},
"node_modules/buffer": {
    "version": "5.7.1",
    "resolved": "https://registry.npmjs.org/buffer/-/buffer-5.7.1.tgz",
    "integrity": "sha512-EHcyIPBQ4BSGlvjB16k5KgAJ27CIsHY/2JBmCRReo48y9rQ3MaUzWX3KVIBa4U7MyX02HdVj0K7C3WaB3ju7FQ==",
    "funding": [
        {
            "type": "github",
            "url": "https://github.com/sponsors/feross"
        }
    ]
}

```

```

},
{
  "type": "patreon",
  "url": "https://www.patreon.com/feross"
},
{
  "type": "consulting",
  "url": "https://feross.org/support"
}
],
"license": "MIT",
"dependencies": {
  "base64-js": "^1.3.1",
  "ieee754": "^1.1.13"
}
},
"node_modules/buffer-crc32": {
  "version": "0.2.13",
  "resolved": "https://registry.npmjs.org/buffer-crc32/-/buffer-crc32-0.2.13.tgz",
  "integrity": "sha512-VO9Ht+p3SN7SKWqcrgEzjGbRSJYTx+Q1pTQC0wrWqHx0vpJraQ6GtHx8tvcg1rlK1byhU5gccxgOgj7B0TDkQ==",
  "license": "MIT",
  "engines": {
    "node": "*"
  },
  "node_modules/buffer-indexof-polyfill": {
    "version": "1.0.2",
    "resolved": "https://registry.npmjs.org/buffer-indexof-polyfill/-/buffer-indexof-polyfill-1.0.2.tgz",
    "integrity": "sha512-I7wzHwA3t1/lwXQh+A5PbNvJxgfo5r3xulgpYDB5zckTu/Z9oUK9biouBKQUjEqzaz3HnAT6TYoovmE+GqSf7A==",
    "license": "MIT",
    "optional": true,
    "engines": {
      "node": ">=0.10"
    }
  },
  "node_modules/buffers": {
    "version": "0.1.1",

```

```

"resolved": "https://registry.npmjs.org/buffers/-/buffers-0.1.1.tgz",
  "integrity": "sha512-9q/rDEGSB/Qsvv2qvzlzdluL5k7AaJOTrw23z9reQthrbF7is4CtlT0DXyO1oei2DCp4uojjzQ7igaSHp1kAEQ==",
    "optional": true,
    "engines": {
      "node": ">=0.2.0"
    },
  },
  "node_modules/bytes": {
    "version": "3.1.2",
    "resolved": "https://registry.npmjs.org/bytes/-/bytes-3.1.2.tgz",
    "integrity": "sha512-/Nf7TyzTx6S3yRJObOAV7956r8cr2+Oj8AC5dt8wSP3BQAoeX58NoHyCU8P8zGkNXStjTSi6fzO6F0pBdcYbEg==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.8"
      },
    },
    "node_modules/call-bind-apply Helpers": {
      "version": "1.0.2",
      "resolved": "https://registry.npmjs.org/call-bind-apply Helpers/-/call-bind-apply Helpers-1.0.2.tgz",
      "integrity": "sha512-Sp1ablJ0ivDkSzjcaJdxEunN5/XvksFJ2sMBFfq6x0ryhQV/2b/KwFe21cMpmHtP0Sij8K99/wSfoEuTObmuMQ==",
        "license": "MIT",
        "dependencies": {
          "es-errors": "^1.3.0",
          "function-bind": "^1.1.2"
        },
      "engines": {
        "node": ">= 0.4"
      },
    },
    "node_modules/call-bound": {
      "version": "1.0.4",
      "resolved": "https://registry.npmjs.org/call-bound/-/call-bound-1.0.4.tgz",
      "integrity": "sha512-+ys997U96po4Kx/ABpBCqhA9EuxJaQWDQg7295H4hBphv3IZg0boBKuwYpt4YXp6MZ5AmZQnU/tyMTIRpaSejg==",
    }
  }
}

```

```

"license": "MIT",
"dependencies": {
  "call-bind-apply Helpers": "^1.0.2",
  "get-intrinsic": "^1.3.0"
},
"engines": {
  "node": ">= 0.4"
},
"funding": {
  "url": "https://github.com/sponsors/ljharb"
}
},
"node_modules/camelcase": {
  "version": "5.3.1",
  "resolved": "https://registry.npmjs.org/camelcase/-/camelcase-5.3.1.tgz",
  "integrity": "sha512-L28STB170nwWS63UjtIEOE3dIdQApAJXZkOI1uMFzf3rRuPegHaHesYee+YxQ+W6SvRDQV6UrdOdRiR153wJg==",
  "license": "MIT",
  "engines": {
    "node": ">=6"
  }
},
"node_modules/chainsaw": {
  "version": "0.1.0",
  "resolved": "https://registry.npmjs.org/chainsaw/-/chainsaw-0.1.0.tgz",
  "integrity": "sha512-75kWfWt6MEKNC8xYXldRpDehRYY/tNSgwKaJq+dbbDcxORuVrrQ+SEHoWsniVn9XPYfP4gmdWleDk/4YNp1rNQ==",
  "license": "MIT/X11",
  "optional": true,
  "dependencies": {
    "traverse": ">=0.3.0 <0.4"
  },
  "engines": {
    "node": "*"
  }
},
"node_modules/chownr": {
  "version": "1.1.4",
  "resolved": "https://registry.npmjs.org/chownr/-/chownr-1.1.4.tgz",

```

```

    "integrity": "sha512-
jJ0bqzaylmJtVnNgzTeSOs8DPavpbYgEr/b0YL8/2GO3xJEhInFmhKMUnEJQjZ
umK7KXGFhUy89PrsJWlakBVg==",
    "license": "ISC"
},
"node_modules/cliui": {
    "version": "6.0.0",
    "resolved": "https://registry.npmjs.org/cliui/-/cliui-6.0.0.tgz",
    "integrity": "sha512-
t6wbgtCXvAzst7QgXxJYqPt0usEfbgQdftEPbLL/cvv6HPE5VgvqCuAIDR0NgU
52ds6rFwqrgakNLrHEjCbrQ==",
    "license": "ISC",
    "dependencies": {
        "string-width": "^4.2.0",
        "strip-ansi": "^6.0.0",
        "wrap-ansi": "^6.2.0"
    }
},
"node_modules/color-convert": {
    "version": "2.0.1",
    "resolved": "https://registry.npmjs.org/color-convert/-/color-convert-2.0.1.tgz",
    "integrity": "sha512-
RRECPsj7iu/xb5oKYcsFHSppFNnsj/52OVTRKb4zP5onXwVF3zVmmToNcOfG
C+CRDpfK/U584fMg38ZHCaEIKQ==",
    "license": "MIT",
    "dependencies": {
        "color-name": "~1.1.4"
    },
    "engines": {
        "node": ">=7.0.0"
    }
},
"node_modules/color-name": {
    "version": "1.1.4",
    "resolved": "https://registry.npmjs.org/color-name/-/color-name-1.1.4.tgz",
    "integrity": "sha512-
dOy+3AuW3a2wNbZHIuMZpTcgjGuLU/uBL/ubcZF9OXbDo8ff4O8yVp5Bf0efS
8uEoYo5q4Fx7dY9OgQGXgAsQA==",
    "license": "MIT"
},
"node_modules/compress-commons": {

```

```

"version": "4.1.2",
  "resolved": "https://registry.npmjs.org/compress-commons/-/compress-commons-4.1.2.tgz",
    "integrity": "sha512-D3uMHTGc/fcO1Gt1/L7i1e33VOvD4A9hfQLP+6ewd+BvG/gQ84Yh4oftEhAdjSMgBgwGL+jsppT7JYNpo6MHHg==",
    "license": "MIT",
    "optional": true,
    "dependencies": {
      "buffer-crc32": "^0.2.13",
      "crc32-stream": "^4.0.2",
      "normalize-path": "^3.0.0",
      "readable-stream": "^3.6.0"
    },
    "engines": {
      "node": ">= 10"
    }
  },
  "node_modules/concat-map": {
    "version": "0.0.1",
    "resolved": "https://registry.npmjs.org/concat-map/-/concat-map-0.0.1.tgz",
    "integrity": "sha512-Srv4dsywQNBfohGpz9o6Yb3Gz3SrUDqBH5rTuhGR7ahtlbYKnVxw2bCFMRIjaA7EXHaXZ8wsHdodFvbkhKmqg==",
    "license": "MIT"
  },
  "node_modules/content-disposition": {
    "version": "0.5.4",
    "resolved": "https://registry.npmjs.org/content-disposition/-/content-disposition-0.5.4.tgz",
    "integrity": "sha512-FveZTNuGw04cxIAiWbzi6zTAL/lhehaWbTtgiuJh4/E95DqMwTmha3KZN1aAWA8cFIhHzMZUvLevkw5Rqk+tSQ==",
    "license": "MIT",
    "dependencies": {
      "safe-buffer": "5.2.1"
    },
    "engines": {
      "node": ">= 0.6"
    }
  },
  "node_modules/content-type": {

```

```

    "version": "1.0.5",
    "resolved": "https://registry.npmjs.org/content-type/-/content-type-1.0.5.tgz",
    "integrity": "sha512-nTjqfcBFEipKdXCv4YDQWCfmcLZKm81IdF0pAopTvyrFGVbcR6P/VAAAd5G7N+0tTr8QqiU0tFadD6FK4NtJwOA==",
    "license": "MIT",
    "engines": {
      "node": ">= 0.6"
    },
    "node_modules/cookie": {
      "version": "0.7.1",
      "resolved": "https://registry.npmjs.org/cookie/-/cookie-0.7.1.tgz",
      "integrity": "sha512-6DnInpx7SJ2AK3+CTUE/ZM0vWTUboZCegxhC2xilydHR9jNuTAASBrfEpHhiGOZw/nX51bHt6YQI8jsGo4y/0w==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/cookie-signature": {
      "version": "1.0.6",
      "resolved": "https://registry.npmjs.org/cookie-signature/-/cookie-signature-1.0.6.tgz",
      "integrity": "sha512-QADzlaHc8icV8I7vbaJXJwod9HWYp8uCqf1xa4OfNu1T7JVxQlrUgOWtHdNDtPiywmFbiS12VjotIXLrKM3orQ==",
      "license": "MIT"
    },
    "node_modules/core-util-is": {
      "version": "1.0.3",
      "resolved": "https://registry.npmjs.org/core-util-is/-/core-util-is-1.0.3.tgz",
      "integrity": "sha512-ZQBvi1DcpJ4GDqanjucZ2Hj3wEO5pZDS89BWbkcrvdksJorwUDDZamX9ldFkp9aw2ImBDLgkObEA4DWNJ9FYQ==",
      "license": "MIT",
      "optional": true
    },
    "node_modules/cors": {
      "version": "2.8.5",
      "resolved": "https://registry.npmjs.org/cors/-/cors-2.8.5.tgz",

```

```

  "integrity": "sha512-
  KIHbLJqu73RGr/hnbrO9uBeixNGuvSQjul/jdFvS/KFSIH1hWVd1ng7zOHx+YrEf
  InLG7q4n6GHQ9cDtxv/P6g==",
  "license": "MIT",
  "dependencies": {
    "object-assign": "^4",
    "vary": "^1"
  },
  "engines": {
    "node": ">= 0.10"
  },
},
"node_modules/crc-32": {
  "version": "1.2.2",
  "resolved": "https://registry.npmjs.org/crc-32/-/crc-32-1.2.2.tgz",
  "integrity": "sha512-
  ROmzCKrTnOwybPcJApAA6WBWij23HVfGVNKqqrZpuyZOHQk2CwHSvpGuy
  t/UNNvaljEd8X5IFGp4Mh+le1IHJQ==",
  "license": "Apache-2.0",
  "optional": true,
  "bin": {
    "crc32": "bin/crc32.njs"
  },
  "engines": {
    "node": ">=0.8"
  },
},
"node_modules/crc32-stream": {
  "version": "4.0.3",
  "resolved": "https://registry.npmjs.org/crc32-stream/-/crc32-stream-
  4.0.3.tgz",
  "integrity": "sha512-
  NT7w2JVU7DFroFdYkeq8cywxrgjPHWkdX1wjpRQXPX5Asews3tA+Ght6lddQ
  O5Mkumffp3X7GEqku3epj2tolw==",
  "license": "MIT",
  "optional": true,
  "dependencies": {
    "crc-32": "^1.2.0",
    "readable-stream": "^3.4.0"
  },
  "engines": {
    "node": ">= 10"
  }
}

```

```

    },
    "node_modules/cross-fetch": {
      "version": "3.1.5",
      "resolved": "https://registry.npmjs.org/cross-fetch/-/cross-fetch-3.1.5.tgz",
      "integrity": "sha512-
lvb1SBsl0Z7GDwmuid+mU3kWVBwTVUbe7S0H52yaaAdQOXq2YktTCZdlAc
NKFzE6QtRz0snpw9bNiPeOlkkQvw==",
      "license": "MIT",
      "dependencies": {
        "node-fetch": "2.6.7"
      }
    },
    "node_modules/cross-fetch/node_modules/node-fetch": {
      "version": "2.6.7",
      "resolved": "https://registry.npmjs.org/node-fetch/-/node-fetch-2.6.7.tgz",
      "integrity": "sha512-
ZjMPFEfVx5j+y2yF35Kzx5sF7kDzxuDj6ziH4FFbOp87zKDZNx8yExJlb05OGF
4Nlt9IHFIMBkRI41VdvcNdbQ==",
      "license": "MIT",
      "dependencies": {
        "whatwg-url": "^5.0.0"
      },
      "engines": {
        "node": "4.x || >=6.0.0"
      },
      "peerDependencies": {
        "encoding": "^0.1.0"
      },
      "peerDependenciesMeta": {
        "encoding": {
          "optional": true
        }
      }
    },
    "node_modules/debug": {
      "version": "2.6.9",
      "resolved": "https://registry.npmjs.org/debug/-/debug-2.6.9.tgz",
      "integrity": "sha512-
bC7ElrdJaJnPbAP+1EotYvqZsb3ecl5wi6Bfi6BJTUCNowp6cvspg0jXznRTKDjm
/E7AdgFBVeAPVMNcKGsHMA==",
      "license": "MIT",
    }
  }
}

```

```

"dependencies": {
  "ms": "2.0.0"
},
"node_modules/decamelize": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/decamelize/-/decamelize-1.2.0.tgz",
  "integrity": "sha512-z2S+W9X73hAUUki+N+9Za2IBlun89zigOyGrsax+KUQ6wKW4ZoWpEYBkGhQjwAjjDCkWxhY0VKEhk8wzY7F5cA==",
  "license": "MIT",
  "engines": {
    "node": ">=0.10.0"
  }
},
"node_modules/depd": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/depd/-/depd-2.0.0.tgz",
  "integrity": "sha512-g7nH6P6dyDioJogAAGprGpCtVlmJhpPk/roCzdb3flh61/s/nPsfR6onyMwkCAR/OIC3yBC0IESvUoQEAsslw==",
  "license": "MIT",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/destroy": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/destroy/-/destroy-1.2.0.tgz",
  "integrity": "sha512-2sJGJTaXIIaR1w4iJSNoN0hnMY7Gpc/n8D4qSCJw8QqFWXf7cuAgnEHxBpw eaVcPevC2l3KpjYCx3NypQQgaJg==",
  "license": "MIT",
  "engines": {
    "node": ">= 0.8",
    "npm": "1.2.8000 || >= 1.4.16"
  }
},
"node_modules/devtools-protocol": {
  "version": "0.0.1045489",
  "resolved": "https://registry.npmjs.org/devtools-protocol/-/devtools-protocol-0.0.1045489.tgz",

```

```

    "integrity": "sha512-
D+PTmWulkuQW4D1NTiCRCFx7pQPn0hgp4YyX4wAQ6xYXKOadSWPR3E
NGDQ47MW/Ewc9v2rpC/UEEGahgBYpSQ==",
    "license": "BSD-3-Clause"
},
"node_modules/dijkstrajs": {
    "version": "1.0.3",
    "resolved": "https://registry.npmjs.org/dijkstrajs/-/dijkstrajs-1.0.3.tgz",
    "integrity": "sha512-
qiSlmBq9+BCdCA/L46dw8Uy93mloxsPSbwnm5yrKn2vMPiy8KyAskTF6zuV/j5
BMsmOGZDPs7KjU+mjb670kfA==",
    "license": "MIT"
},
"node_modules/dotenv": {
    "version": "16.6.1",
    "resolved": "https://registry.npmjs.org/dotenv/-/dotenv-16.6.1.tgz",
    "integrity": "sha512-
uBq4egWHTcTt33a72vpSG0z3HnPul6NqYcTrKEg2azoEyl2hpW0zqlxysq2pK
9HIDIHyakeYaYnSAwd8bow==",
    "license": "BSD-2-Clause",
    "engines": {
        "node": ">=12"
    },
    "funding": {
        "url": "https://dotenvx.com"
    }
},
"node_modules/dunder-proto": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/dunder-proto/-/dunder-proto-1.0.1.tgz",
    "integrity": "sha512-
KIN/nDJBQRcXw0MLVhZE9iQHmG68qAVIBg9CqmUYjmQlhgij9U5MFvrqkUL
5FbtyyzZuOeOt0zdeRe4UY7ct+A==",
    "license": "MIT",
    "dependencies": {
        "call-bind-apply Helpers": "^1.0.1",
        "es-errors": "^1.3.0",
        "gopd": "^1.2.0"
    },
    "engines": {
        "node": ">= 0.4"
    }
}

```

```

    },
  },
  "node_modules/duplexer2": {
    "version": "0.1.4",
    "resolved": "https://registry.npmjs.org/duplexer2/-/duplexer2-0.1.4.tgz",
    "integrity": "sha512-asLFVfWWtJ90ZyOuHMqk7/S2w2guQKxUI2itj3d92ADHhxUSbCMGi1f1cBcJ7xM1To+pE/Khbwo1yuNbMEPKeA==",
    "license": "BSD-3-Clause",
    "optional": true,
    "dependencies": {
      "readable-stream": "^2.0.2"
    }
  },
  "node_modules/duplexer2/node_modules/readable-stream": {
    "version": "2.3.8",
    "resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-2.3.8.tgz",
    "integrity": "sha512-8p0AUk4XODglewSi0I8Epjs+EVnWiK7NoDIEGU0HhE7+ZyY8D1IMY7odu5lRrFXGg71L15KG8QrPmum45RTtdA==",
    "license": "MIT",
    "optional": true,
    "dependencies": {
      "core-util-is": "~1.0.0",
      "inherits": "~2.0.3",
      "isarray": "~1.0.0",
      "process-nextick-args": "~2.0.0",
      "safe-buffer": "~5.1.1",
      "string_decoder": "~1.1.1",
      "util-deprecate": "~1.0.1"
    }
  },
  "node_modules/duplexer2/node_modules/safe-buffer": {
    "version": "5.1.2",
    "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.2.tgz",
    "integrity": "sha512-Gd2UZBJDkXIY7GbJxfsE8/nvKkUEU1G38c1siN6QP6a9PT9MmHB8GnpSCSmMJSoF8LOlrt8ud/wPtojys4G6+g==",
    "license": "MIT",
    "optional": true
  },

```

```

"node_modules/duplexer2/node_modules/string_decoder": {
  "version": "1.1.1",
  "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-1.1.1.tgz",
  "integrity": "sha512-n/ShnDi6FHbbVfviro+WojiFzv+s8MPMHBCzVePfUpDJLwoLT0ht1I4YwBCbi8pJAveEEdnkHyPyTP/mzRfwg==",
  "license": "MIT",
  "optional": true,
  "dependencies": {
    "safe-buffer": "~5.1.0"
  }
},
"node_modules/ee-first": {
  "version": "1.1.1",
  "resolved": "https://registry.npmjs.org/ee-first/-/ee-first-1.1.1.tgz",
  "integrity": "sha512-WMwm9LhRUo+WUaRN+vRuETqG89lgZphVSNkdFgeb6sS/E4OrDIN7t48CAewSHXc6C8lefD8KKfr5vY61brQlow==",
  "license": "MIT"
},
"node_modules/emoji-regex": {
  "version": "8.0.0",
  "resolved": "https://registry.npmjs.org/emoji-regex/-/emoji-regex-8.0.0.tgz",
  "integrity": "sha512-MSjYzcWNOA0ewAHpz0MxpYFvwg6yjy1NG3xteoqz644VCo/RPgnr1/GGt+ic3iJTzQ8Eu3TdM14SawnVUmGE6A==",
  "license": "MIT"
},
"node_modules/encodeurl": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-2.0.0.tgz",
  "integrity": "sha512-Q0n9HRi4m6JuGIV1eFlmvJB7ZEVxu93lrMyiMsGC0lrMJMWzRgx6WGquyfQgZVb31vhGgXnfmPNNXmxnOkRBrg==",
  "license": "MIT",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/end-of-stream": {
  "version": "1.4.5",

```

```

  "resolved": "https://registry.npmjs.org/end-of-stream/-/end-of-stream-1.4.5.tgz",
    "integrity": "sha512-ooEGc6HP26xXq/N+GCGOT0JKCLDGrq2bQUZrQ7gyrJiZANJ/8YDTxTpQBXGMn+WbIQXNVpyWymm7KYVICQnyOg==",
    "license": "MIT",
    "dependencies": {
      "once": "^1.4.0"
    }
  },
  "node_modules/es-define-property": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/es-define-property/-/es-define-property-1.0.1.tgz",
      "integrity": "sha512-e3nRfgfUZ4rNGL232gUgX06QNyyez04KdjFrF+LTRoOXmrOgFKDg4BCdsjW8EnT69eqdYGmRpJwiPVYNrCaW3g==",
      "license": "MIT",
      "engines": {
        "node": ">= 0.4"
      }
    },
    "node_modules/es-errors": {
      "version": "1.3.0",
      "resolved": "https://registry.npmjs.org/es-errors/-/es-errors-1.3.0.tgz",
        "integrity": "sha512-Zf5H2Kxt2xjTvbJvP2ZWLEICxA6j+hAmMzllypy4xcBg1vKVnx89Wy0GbS+kf5cwCVFFzdCFh2XSCFNULS6csw==",
        "license": "MIT",
        "engines": {
          "node": ">= 0.4"
        }
      },
      "node_modules/es-object-atoms": {
        "version": "1.1.1",
        "resolved": "https://registry.npmjs.org/es-object-atoms/-/es-object-atoms-1.1.1.tgz",
          "integrity": "sha512-FGgH2h8zKNim9Ijj7dankFPcICIK9Cp5bm+c2gQSYePhpaG5+esrLODihlorn+Pe6FGJzWhXQotPv73jTaldXA==",
          "license": "MIT",
          "dependencies": {

```

```

    "es-errors": "^1.3.0"
},
"engines": {
    "node": ">= 0.4"
}
},
"node_modules/escalade": {
    "version": "3.2.0",
    "resolved": "https://registry.npmjs.org/escalade/-/escalade-3.2.0.tgz",
    "integrity": "sha512-WUj2qlxaQtO4g6Pq5c29GTcWGDyd8itL8zTlipgECz3JesAiiOKotd8JU6otB3PA
CgG6xkJUyVhboMS+bje/jA==",
    "license": "MIT",
    "engines": {
        "node": ">=6"
    }
},
"node_modules/escape-html": {
    "version": "1.0.3",
    "resolved": "https://registry.npmjs.org/escape-html/-/escape-html-1.0.3.tgz",
    "integrity": "sha512-NiSupZ4OeuGwr68lGleym/ksIZMJodUGOSCZ/FSnTxcrekbvqrgdUxIJOMpijaK
ZVjAJrWrGs/6Jy8OMuj9ow==",
    "license": "MIT"
},
"node_modules/etag": {
    "version": "1.8.1",
    "resolved": "https://registry.npmjs.org/etag/-/etag-1.8.1.tgz",
    "integrity": "sha512-aIL5Fx7mawVa300al2BnEE4iNvo1qETxLrPI/o05L7z6go7fCw1J6EQmbK4FmJ
2AS7kgVF/KEZWufBfdCIMcPg==",
    "license": "MIT",
    "engines": {
        "node": ">= 0.6"
    }
},
"node_modules/express": {
    "version": "4.21.2",
    "resolved": "https://registry.npmjs.org/express/-/express-4.21.2.tgz",
    "integrity": "sha512-28HqgMZAmih1Czt9ny7qr6ek2qddF4FcIbMzwhCREB6OFFH+rXAnuNCwo1/w
FvrtbgsQDb4kSbX9de9lFbrXnA==",

```

```
"license": "MIT",
"dependencies": {
  "accepts": "~1.3.8",
  "array-flatten": "1.1.1",
  "body-parser": "1.20.3",
  "content-disposition": "0.5.4",
  "content-type": "~1.0.4",
  "cookie": "0.7.1",
  "cookie-signature": "1.0.6",
  "debug": "2.6.9",
  "depd": "2.0.0",
  "encodeurl": "~2.0.0",
  "escape-html": "~1.0.3",
  "etag": "~1.8.1",
  "finalhandler": "1.3.1",
  "fresh": "0.5.2",
  "http-errors": "2.0.0",
  "merge-descriptors": "1.0.3",
  "methods": "~1.1.2",
  "on-finished": "2.4.1",
  "parseurl": "~1.3.3",
  "path-to-regexp": "0.1.12",
  "proxy-addr": "~2.0.7",
  "qs": "6.13.0",
  "range-parser": "~1.2.1",
  "safe-buffer": "5.2.1",
  "send": "0.19.0",
  "serve-static": "1.16.2",
  "setprototypeof": "1.2.0",
  "statuses": "2.0.1",
  "type-is": "~1.6.18",
  "utils-merge": "1.0.1",
  "vary": "~1.1.2"
},
"engines": {
  "node": ">= 0.10.0"
},
"funding": {
  "type": "opencollective",
  "url": "https://opencollective.com/express"
}
},
```

```

"node_modules/extract-zip": {
  "version": "2.0.1",
  "resolved": "https://registry.npmjs.org/extract-zip/-/extract-zip-2.0.1.tgz",
  "integrity": "sha512-GDhU9ntwuKyGXdZBUgTle+vXnWj0fppUEtMDL0+idd5Sta8TGpHssn/eusA9mrPr9qNDym6SxAYZjNvCn/9RBg==",
  "license": "BSD-2-Clause",
  "dependencies": {
    "debug": "^4.1.1",
    "get-stream": "^5.1.0",
    "yauzl": "^2.10.0"
  },
  "bin": {
    "extract-zip": "cli.js"
  },
  "engines": {
    "node": ">= 10.17.0"
  },
  "optionalDependencies": {
    "@types/yauzl": "^2.9.1"
  }
},
"node_modules/extract-zip/node_modules/debug": {
  "version": "4.4.1",
  "resolved": "https://registry.npmjs.org/debug/-/debug-4.4.1.tgz",
  "integrity": "sha512-KcKCqiftBJcZr++7ykoDIEwSa3XWowTfNPo92BYxjXiyYEvrUQh2aLyhxBCwww+heortUFxEJYcRzosstTEBYQ==",
  "license": "MIT",
  "dependencies": {
    "ms": "^2.1.3"
  },
  "engines": {
    "node": ">=6.0"
  },
  "peerDependenciesMeta": {
    "supports-color": {
      "optional": true
    }
  }
},
"node_modules/extract-zip/node_modules/ms": {

```

```

    "version": "2.1.3",
    "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
    "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTIA==",
    "license": "MIT"
},
"node_modules/fd-slicer": {
    "version": "1.1.0",
    "resolved": "https://registry.npmjs.org/fd-slicer/-/fd-slicer-1.1.0.tgz",
    "integrity": "sha512-cE1qsB/VwyQozZ+q1dGxR8LBYNZeofhEdUNGSMbQD3Gw2IAzX9Zb3uIU6Eb/Fmyjo9AWWfnn0AUCHqtevs/8g==",
    "license": "MIT",
    "dependencies": {
        "pend": "~1.2.0"
    }
},
"node_modules/finalhandler": {
    "version": "1.3.1",
    "resolved": "https://registry.npmjs.org/finalhandler/-/finalhandler-1.3.1.tgz",
    "integrity": "sha512-6BN9trH7bp3qvnrRyzsBz+g3IZxTNZTbVO2EV1CS0WIcDbawYVdYvGfIME/9QP0h0pYICDBCTjYa9nZzMDpyxQ==",
    "license": "MIT",
    "dependencies": {
        "debug": "2.6.9",
        "encodeurl": "~2.0.0",
        "escape-html": "~1.0.3",
        "on-finished": "2.4.1",
        "parseurl": "~1.3.3",
        "statuses": "2.0.1",
        "unpipe": "~1.0.0"
    }
},
"engines": {
    "node": ">= 0.8"
}
},
"node_modules/find-up": {
    "version": "4.1.0",
    "resolved": "https://registry.npmjs.org/find-up/-/find-up-4.1.0.tgz",

```

```

  "integrity": "sha512-PpOwAdQ/YIXQ2vj8a3h8lipDuYRi3wceVQQGYWxNINccq40Anw7BlsEXCMbt1Zt+OLA6Fq9sulpIWD0OsnISlw==",
  "license": "MIT",
  "dependencies": {
    "locate-path": "^5.0.0",
    "path-exists": "^4.0.0"
  },
  "engines": {
    "node": ">=8"
  },
  "node_modules/fluent-ffmpeg": {
    "version": "2.1.3",
    "resolved": "https://registry.npmjs.org/fluent-ffmpeg/-/fluent-ffmpeg-2.1.3.tgz",
    "integrity": "sha512-Be3narBNt2s6bsaqP6Jzq91heDgOEaDCJAXcE3qcma/EJBSy5FB4cvO31XBInuAuKBx8Kptf8dkhjK0lOru39Q==",
    "deprecated": "Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.",
    "license": "MIT",
    "dependencies": {
      "async": "^0.2.9",
      "which": "^1.1.1"
    },
    "engines": {
      "node": ">=18"
    }
  },
  "node_modules/fluent-ffmpeg/node_modules/async": {
    "version": "0.2.10",
    "resolved": "https://registry.npmjs.org/async/-/async-0.2.10.tgz",
    "integrity": "sha512-eAkdoKxU6/LkKDBzLpT+t6Ff5EtSF4wx1WfJiPEEV7WNLnDaRXk0oVysiEPm262roaachGexwUv94WhSgN5TQ=="
  },
  "node_modules/forwarded": {
    "version": "0.2.0",
    "resolved": "https://registry.npmjs.org/forwarded/-/forwarded-0.2.0.tgz",

```

```

  "integrity": "sha512-
buRG0fpBtRHSTCOASe6hD258tEubFoRLb4ZNA6NxMVHNw2gOcwHo9wyabl
zMzOA5z9xA9L1KNjk/Nt6MT9aYow==",
  "license": "MIT",
  "engines": {
    "node": ">= 0.6"
  },
  "node_modules/fresh": {
    "version": "0.5.2",
    "resolved": "https://registry.npmjs.org/fresh/-/fresh-0.5.2.tgz",
    "integrity": "sha512-
zJ2mQYM18rEF0udeV4GShTGIQ7RbzA7ozbU9I/XBpm7kqgMywgmyIMwXHx
ZJmkVoYkna9d2pVXVXPdYTP9ej8Q==",
    "license": "MIT",
    "engines": {
      "node": ">= 0.6"
    },
    "node_modules/fs-constants": {
      "version": "1.0.0",
      "resolved": "https://registry.npmjs.org/fs-constants/-/fs-constants-1.0.0.tgz",
      "integrity": "sha512-
y6OAwoSIf7FyjMlv94u+b5rdheZEjzR63GTyZJm5qh4Bi+2YgwLCcl/fPFZkL5PS
ixOt6ZNKm+w+Hfp/Bciwow==",
      "license": "MIT"
    },
    "node_modules/fs-extra": {
      "version": "10.1.0",
      "resolved": "https://registry.npmjs.org/fs-extra/-/fs-extra-10.1.0.tgz",
      "integrity": "sha512-
oRXApq54ETRj4eMiFzGnHWGy+zo5raudjuxN0b8H7s/RU2oW0Wvsx9O0ACRN/kRq9E8Vu/ReskGB5o3ji+FzHQ==",
      "license": "MIT",
      "optional": true,
      "dependencies": {
        "graceful-fs": "^4.2.0",
        "jsonfile": "^6.0.1",
        "universalify": "^2.0.0"
      },
      "engines": {
        "node": ">=12"
      }
    }
  }
}

```

```

    },
    "node_modules/fs.realpath": {
      "version": "1.0.0",
      "resolved": "https://registry.npmjs.org/fs.realpath/-/fs.realpath-1.0.0.tgz",
      "integrity": "sha512-OO0pH2IK6a0hZnAdau5ltzHPI6pUlVl7jMVnxUQRtw4owF2wk8IOSabtGDCTP4Grg2MbGnWO9X8K1t4+fGMDw==",
      "license": "ISC"
    },
    "node_modules/fstream": {
      "version": "1.0.12",
      "resolved": "https://registry.npmjs.org/fstream/-/fstream-1.0.12.tgz",
      "integrity": "sha512-WvJ193OHa0GHPEL+AycEJgqvBEwyfRkN1vhjca23OaPVMCaLCXTd5qAu82AjTcgP1UJmytkOKb63Ypde7raDIg==",
      "deprecated": "This package is no longer supported.",
      "license": "ISC",
      "optional": true,
      "dependencies": {
        "graceful-fs": "^4.1.2",
        "inherits": "~2.0.0",
        "mkdirp": ">=0.5 0",
        "rimraf": "2"
      },
      "engines": {
        "node": ">=0.6"
      }
    },
    "node_modules/fstream/node_modules/rimraf": {
      "version": "2.7.1",
      "resolved": "https://registry.npmjs.org/rimraf/-/rimraf-2.7.1.tgz",
      "integrity": "sha512-uWjbAKK3T1OSVptzX7NI6PvQ3qAGtKEtVRjRuazjfL3Bx5el409VZSqqND+4UNnmzLVdPj9FqFJNPqBZFve4w==",
      "deprecated": "Rimraf versions prior to v4 are no longer supported",
      "license": "ISC",
      "optional": true,
      "dependencies": {
        "glob": "^7.1.3"
      },
      "bin": {

```

```

    "rimraf": "bin.js"
  }
},
"node_modules/function-bind": {
  "version": "1.1.2",
  "resolved": "https://registry.npmjs.org/function-bind/-/function-bind-1.1.2.tgz",
  "integrity": "sha512-7XHNxH7qX9xG5mlwxkhumTox/MIRNcOgDrxWsMt2pAr23WHp6MrRIN7FBSFpCpr+oVO0F744iUgR82nJMfG2SA==",
  "license": "MIT",
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/get-caller-file": {
  "version": "2.0.5",
  "resolved": "https://registry.npmjs.org/get-caller-file/-/get-caller-file-2.0.5.tgz",
  "integrity": "sha512-DyFP3BM/3YHTQOCUL/w0OZHR0lpKeGrxotcHWcqNEdnltqFwXVfhEBQ94el034AfQpo0rGki4cyliftY06h2Fg==",
  "license": "ISC",
  "engines": {
    "node": "6.* || 8.* || >= 10.*"
  }
},
"node_modules/get-intrinsic": {
  "version": "1.3.0",
  "resolved": "https://registry.npmjs.org/get-intrinsic/-/get-intrinsic-1.3.0.tgz",
  "integrity": "sha512-9fSjSaos/fRIVIp+xSJIE6lfwhES7LNtKaCBlamHsjr2na1BiABJPo0mOjjz8GJDURamCPGqaiVg5mfjb98CQ==",
  "license": "MIT",
  "dependencies": {
    "call-bind-apply Helpers": "^1.0.2",
    "es-define-property": "^1.0.1",
    "es-errors": "^1.3.0",
    "es-object-atoms": "^1.1.1",
    "function-bind": "^1.1.2",
    "get-proto": "^1.0.1",
    "gopd": "^1.2.0",
  }
}

```

```

    "has-symbols": "^1.1.0",
    "hasown": "^2.0.2",
    "math-intrinsics": "^1.1.0"
},
"engines": {
    "node": ">= 0.4"
},
"funding": {
    "url": "https://github.com/sponsors/ljharb"
}
},
"node_modules/get-proto": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/get-proto/-/get-proto-1.0.1.tgz",
    "integrity": "sha512-sTSfBjoXBp89JvIKlefqw7U2CCebSc74kiY6awiGogKtoSGbjYE/G/+I9sF3MWFPNc9lcoOC4ODfKHfxFmp0g==",
    "license": "MIT",
    "dependencies": {
        "dunder-proto": "^1.0.1",
        "es-object-atoms": "^1.0.0"
    },
    "engines": {
        "node": ">= 0.4"
    }
},
"node_modules/get-stream": {
    "version": "5.2.0",
    "resolved": "https://registry.npmjs.org/get-stream/-/get-stream-5.2.0.tgz",
    "integrity": "sha512-nBF+F1rAZVCu/p7rjzgA+Yb4IfYXrpl7a6VmJrU8wF9I1CKvP/QwPNZHnOlwbTkY6dvtFlzFMSyQXbLoTQPRpA==",
    "license": "MIT",
    "dependencies": {
        "pump": "^3.0.0"
    },
    "engines": {
        "node": ">=8"
    },
    "funding": {
        "url": "https://github.com/sponsors/sindresorhus"
    }
}

```

```

},
"node_modules/glob": {
  "version": "7.2.3",
  "resolved": "https://registry.npmjs.org/glob/-/glob-7.2.3.tgz",
  "integrity": "sha512-nFR0zLpU2YCaRxwoCJvL6UvCH2JFyFVIvwTLsIf21AuHIMskA1hhTdk+LIYJtOIJt9v6dvszD2BGRqBL+iQK9Q==",
  "deprecated": "Glob versions prior to v9 are no longer supported",
  "license": "ISC",
  "dependencies": {
    "fs.realpath": "^1.0.0",
    "inflight": "^1.0.4",
    "inherits": "2",
    "minimatch": "^3.1.1",
    "once": "^1.3.0",
    "path-is-absolute": "^1.0.0"
  },
  "engines": {
    "node": "*"
  },
  "funding": {
    "url": "https://github.com/sponsors/isaacs"
  }
},
"node_modules/gopd": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/gopd/-/gopd-1.2.0.tgz",
  "integrity": "sha512-ZUKRh6/kUFoAiTAtTYPZJ3hw9wNxx+BIBOijnIG9PnrJsCcSjs1wyD6vJpaYtg nzDrKYRSqf3OO6Rfa93xsRg==",
  "license": "MIT",
  "engines": {
    "node": ">= 0.4"
  },
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/graceful-fs": {
  "version": "4.2.11",
  "resolved": "https://registry.npmjs.org/graceful-fs/-/graceful-fs-4.2.11.tgz",

```

```

    "integrity": "sha512-
RbJ5/jmFcNNCcDV5o9eTnBLJ/HszWV0P73bc+Ff4nS/rJj+YaS6IGyiOL0VoBY
X+I1Wrl3k63h/KrH+nhJ0XvQ==",
    "license": "ISC",
    "optional": true
},
"node_modules/has-symbols": {
    "version": "1.1.0",
    "resolved": "https://registry.npmjs.org/has-symbols/-/has-symbols-1.1.0.tgz",
    "integrity": "sha512-
1cDNDwJ2Jaohmb3sg4OmKaMBwuC48sYni5HUw2DvsC8LjGTLK9h+eb1X6R
yuOHe4hT0ULCW68iomhjUoKUqlPQ==",
    "license": "MIT",
    "engines": {
        "node": ">= 0.4"
    },
    "funding": {
        "url": "https://github.com/sponsors/ljharb"
    }
},
"node_modules/hasown": {
    "version": "2.0.2",
    "resolved": "https://registry.npmjs.org/hasown/-/hasown-2.0.2.tgz",
    "integrity": "sha512-
0hJU9SCPvmMzIBdZFqNPXWa6dqh7WdH0cll9y+CyS8rG3nL48Bclra9HmKh
VVUHyPWNH5Y7xDwAB7bfgSjkUMQ==",
    "license": "MIT",
    "dependencies": {
        "function-bind": "^1.1.2"
    },
    "engines": {
        "node": ">= 0.4"
    }
},
"node_modules/http-errors": {
    "version": "2.0.0",
    "resolved": "https://registry.npmjs.org/http-errors/-/http-errors-2.0.0.tgz",
    "integrity": "sha512-
FtwrG/euBzaEjYeRqOgly7G0qviiXoJWnvEH2Z1plBdXgbyjv34pHTSb9zoeHMy
Dy33+DWy5Wt9Wo+TURtOYSQ==",
    "license": "MIT",
    "dependencies": {

```

```

    "depd": "2.0.0",
    "inherits": "2.0.4",
    "setPrototypeOf": "1.2.0",
    "statuses": "2.0.1",
    "toIdentifier": "1.0.1"
},
"engines": {
    "node": ">= 0.8"
}
},
"node_modules/https-proxy-agent": {
    "version": "5.0.1",
    "resolved": "https://registry.npmjs.org/https-proxy-agent/-/https-proxy-agent-5.0.1.tgz",
    "integrity": "sha512-dFcAjpTQFgoLMzC2VwU+C/CbS7uRL0IWmxDITmqm7C+7F0Odmj6s9l6alZc6AELXhrnggM2CeWSXHGOdX2YtwA==",
    "license": "MIT",
    "dependencies": {
        "agent-base": "6",
        "debug": "4"
    },
    "engines": {
        "node": ">= 6"
    }
},
"node_modules/https-proxy-agent/node_modules/debug": {
    "version": "4.4.1",
    "resolved": "https://registry.npmjs.org/debug/-/debug-4.4.1.tgz",
    "integrity": "sha512-KcKCqiftBJcZr++7ykoDIewSa3XWowTfNPo92BYxjXiyYEvrUQh2aLyhxBCwww+heortUFxEJYcRzosstTEBYQ==",
    "license": "MIT",
    "dependencies": {
        "ms": "^2.1.3"
    },
    "engines": {
        "node": ">=6.0"
    }
},
"peerDependenciesMeta": {
    "supports-color": {
        "optional": true
    }
}
}

```

```

    }
},
},
"node_modules/https-proxy-agent/node_modules/ms": {
  "version": "2.1.3",
  "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
  "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTIA==",
  "license": "MIT"
},
"node_modules/iconv-lite": {
  "version": "0.4.24",
  "resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.4.24.tgz",
  "integrity": "sha512-v3MXnZAcvnywkTUEZomIActle7RXeedOR31wwl7VlyoXO4Qi9arvSenNQWne1TcRwhCL1HwLI21bEqdpj8/rA==",
  "license": "MIT",
  "dependencies": {
    "safer-buffer": ">= 2.1.2 < 3"
  },
  "engines": {
    "node": ">=0.10.0"
  }
},
"node_modules/ieee754": {
  "version": "1.2.1",
  "resolved": "https://registry.npmjs.org/ieee754/-/ieee754-1.2.1.tgz",
  "integrity": "sha512-dcyqhDvX1C46IXZcVqCpK+FtMRQVdIMN6/Df5js2zouUsqG7I6sFxitIC+7KYK29KdXOLHdu9zL4sFnoVQnqaA==",
  "funding": [
    {
      "type": "github",
      "url": "https://github.com/sponsors/feross"
    },
    {
      "type": "patreon",
      "url": "https://www.patreon.com/feross"
    },
    {
      "type": "consulting",
    }
  ]
}

```

```

        "url": "https://feross.org/support"
    }
],
"license": "BSD-3-Clause"
},
"node_modules/inflight": {
    "version": "1.0.6",
    "resolved": "https://registry.npmjs.org/inflight/-/inflight-1.0.6.tgz",
    "integrity": "sha512-k92I/b08q4wvFscXCLvqfsHCrjrF7yiXsQuIVvVE7N82W3+aqpzuUdBbfhWcy/FZR3/4lgfIMgKLOsvPDrGCJA==",
    "deprecated": "This module is not supported, and leaks memory. Do not use it. Check out Iru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.",
    "license": "ISC",
    "dependencies": {
        "once": "^1.3.0",
        "wrappy": "1"
    }
},
"node_modules/inherits": {
    "version": "2.0.4",
    "resolved": "https://registry.npmjs.org/inherits/-/inherits-2.0.4.tgz",
    "integrity": "sha512-k/vGaX4/Yla3WzyMCvTQOXYelHvqOKtnqBduzTHpzpQZzAskKMhZ2K+EnBiSM9zGSolFeMpXKxa4dYeZIQqewQ==",
    "license": "ISC"
},
"node_modules/ipaddr.js": {
    "version": "1.9.1",
    "resolved": "https://registry.npmjs.org/ipaddr.js/-/ipaddr.js-1.9.1.tgz",
    "integrity": "sha512-OKI/607xoxSToH7GjN1FfSbLoU0+btTicjsQSWQlh/hZykN8KpmMf7uYwPW3R+akZ6R/w18ZIXSHBYXiYUPO3g==",
    "license": "MIT",
    "engines": {
        "node": ">= 0.10"
    }
},
"node_modules/is-fullwidth-code-point": {
    "version": "3.0.0",

```

```

  "resolved": "https://registry.npmjs.org/is-fullwidth-code-point/-/is-fullwidth-code-point-3.0.0.tgz",
  "integrity": "sha512-zymm5+u+sCsSWyD9qNaejV3DFvhCKclKdizYaJUuHA83RLjb7nSuGnndCHGv0hk+KY7BMAIsWeK4Ueg6EV6XQg==",
  "license": "MIT",
  "engines": {
    "node": ">=8"
  },
  "node_modules/isarray": {
    "version": "1.0.0",
    "resolved": "https://registry.npmjs.org/isarray/-/isarray-1.0.0.tgz",
    "integrity": "sha512-VLghIWm6ELQzo7zwmcg0NmTVyWKYjvleM83yjp0wRDTmUnrM678fQbcKBo6n2CJEF0sz0G//ytg+TKla89ALQ==",
    "license": "MIT",
    "optional": true
  },
  "node_modules/isexe": {
    "version": "2.0.0",
    "resolved": "https://registry.npmjs.org/isexe/-/isexe-2.0.0.tgz",
    "integrity": "sha512-RHxMLp9lnKHGHRng9QFhRCMbYAcVpn69smSGcq3f36xjgVVWThj4qqLbTLIq7Ssj8B+fIQ1EuCEGI2IKsyQelw==",
    "license": "ISC"
  },
  "node_modules/jsonfile": {
    "version": "6.1.0",
    "resolved": "https://registry.npmjs.org/jsonfile/-/jsonfile-6.1.0.tgz",
    "integrity": "sha512-5dgndWOrIYSm5cnYaJNhailNDKOqFwyDB/rr1E9ZsGciGvKPs8R2xYGCacuf3z6K1YKDz182fd+fY3cn3pMqXQ==",
    "license": "MIT",
    "optional": true,
    "dependencies": {
      "universalify": "^2.0.0"
    },
    "optionalDependencies": {
      "graceful-fs": "^4.1.6"
    }
  },

```

```

"node_modules/lazystream": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/lazystream/-/lazystream-1.0.1.tgz",
  "integrity": "sha512-b94GiNHQNy6JNTrt5w6zNyffMrNkXZb3KTkCZJb2V1xaEGCk093vkZ2jk3tpaeP33/OiXC+WvK9AxUebnf5nbw==",
  "license": "MIT",
  "optional": true,
  "dependencies": {
    "readable-stream": "^2.0.5"
  },
  "engines": {
    "node": ">= 0.6.3"
  }
},
"node_modules/lazystream/node_modules/readable-stream": {
  "version": "2.3.8",
  "resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-2.3.8.tgz",
  "integrity": "sha512-8p0AUk4XODglewSi0I8Epjs+EVnWiK7NoDIEGU0HhE7+ZyY8D1IMY7odu5IRrFXGg71L15KG8QrPnum45RTtdA==",
  "license": "MIT",
  "optional": true,
  "dependencies": {
    "core-util-is": "~1.0.0",
    "inherits": "~2.0.3",
    "isarray": "~1.0.0",
    "process-nextick-args": "~2.0.0",
    "safe-buffer": "~5.1.1",
    "string_decoder": "~1.1.1",
    "util-deprecate": "~1.0.1"
  }
},
"node_modules/lazystream/node_modules/safe-buffer": {
  "version": "5.1.2",
  "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.2.tgz",
  "integrity": "sha512-Gd2UZBJDkXIY7GbJxfsE8/nvKkUEU1G38c1siN6QP6a9PT9MmHB8GnpSCSMJSoF8LOlrt8ud/wPtojys4G6+g==",
  "license": "MIT",
  "optional": true
}

```

```

},
"node_modules/lazystream/node_modules/string_decoder": {
  "version": "1.1.1",
  "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-1.1.1.tgz",
  "integrity": "sha512-n/ShnvDi6FHbbVfviro+WojiFzv+s8MPMHBCzVePfUpDJLwoLT0ht1I4YwBCbi8pJAveEEdnkHyPyTP/mzRfwg==",
  "license": "MIT",
  "optional": true,
  "dependencies": {
    "safe-buffer": "~5.1.0"
  }
},
"node_modules/listenercount": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/listenercount/-/listenercount-1.0.1.tgz",
  "integrity": "sha512-3mk/Zag0+IJxeDrxSgaDPy4zZ3w05PRZeJNnIWhzFz5OkX49J4krc+A8X2d2M69vGMBEX0uyI8M+W+8gH+kBqQ==",
  "license": "ISC",
  "optional": true
},
"node_modules/locate-path": {
  "version": "5.0.0",
  "resolved": "https://registry.npmjs.org/locate-path/-/locate-path-5.0.0.tgz",
  "integrity": "sha512-t7hw9pI+WvuwNJXwk5zVHpyhlqzg2qTlklJOf0mVxGSbe3Fp2VieZcdunYjaLDo y6p9uGpQEGWG87WpMKINq8g==",
  "license": "MIT",
  "dependencies": {
    "p-locate": "^4.1.0"
  },
  "engines": {
    "node": ">=8"
  }
},
"node_modules/lodash.defaults": {
  "version": "4.2.0",
  "resolved": "https://registry.npmjs.org/lodash.defaults/-/lodash.defaults-4.2.0.tgz",

```

```

    "integrity": "sha512-qjxPLHd3r5DnsdGacqOMU6pb/avJzdh9tFX2ymgoZE27BmjXrNy/y4LoaiTeAb+O3gL8AfpJGtqfX/ae2leYYQ==",
    "license": "MIT",
    "optional": true
},
"node_modules/lodash.difference": {
    "version": "4.5.0",
    "resolved": "https://registry.npmjs.org/lodash.difference/-/lodash.difference-4.5.0.tgz",
    "integrity": "sha512-dS2j+W26TQ7taQBGN8Lbbq04ssV3emRw4NY58WErlTO29plqS0HmoT5aJ9+TUQ1N3G+JOZSji4eugsWwGp9yPA==",
    "license": "MIT",
    "optional": true
},
"node_modules/lodash.flatten": {
    "version": "4.4.0",
    "resolved": "https://registry.npmjs.org/lodash.flatten/-/lodash.flatten-4.4.0.tgz",
    "integrity": "sha512-C5N2Z3DggnKr0LOpv/hKCgKdb7ZZwaflrsesve6lmzvZIRZRGaZ/l6Q8+2W7NaT+ZwO3fFISCzCzrDCFdJfZ4g==",
    "license": "MIT",
    "optional": true
},
"node_modules/lodash.isplainobject": {
    "version": "4.0.6",
    "resolved": "https://registry.npmjs.org/lodash.isplainobject/-/lodash.isplainobject-4.0.6.tgz",
    "integrity": "sha512-oSXzaWypCMHkPC3NvBEaPHf0KsA5mvPrOPgQWDsbg8n7orZ290M0BmC/jgRZ4vcJ6DTAhjrsSYgdsW/F+MFOBA==",
    "license": "MIT",
    "optional": true
},
"node_modules/lodash.union": {
    "version": "4.6.0",
    "resolved": "https://registry.npmjs.org/lodash.union/-/lodash.union-4.6.0.tgz",

```

```

    "integrity": "sha512-c4pB2CdGrGdjMKYLA+XiRDO7Y0PRQbm/Gzg8qMj+QH+pFVAoTp5sBpO0odL3FjoPCGjK96p6qsP+yQoiLoOBcw==",
    "license": "MIT",
    "optional": true
},
"node_modules/math-intrinsics": {
    "version": "1.1.0",
    "resolved": "https://registry.npmjs.org/math-intrinsics/-/math-intrinsics-1.1.0.tgz",
    "integrity": "sha512-IXtbwEk5HTPyEwyKX6hGkYXxM9nbj64B+iVJnC/R6B0pH5G4V3b0pVbL7DBj4tkhBAAppbQUlf6F6XI9LHu1g==",
    "license": "MIT",
    "engines": {
        "node": ">= 0.4"
    }
},
"node_modules/media-typer": {
    "version": "0.3.0",
    "resolved": "https://registry.npmjs.org/media-typer/-/media-typer-0.3.0.tgz",
    "integrity": "sha512-dq+qelQ9akHpcOl/gUVRTxVIOkAJ1wR3QAvb4RsVjS8oVoFjDGTc679wJYmUmknUF5HwMLOb5O+a3KxfWapPQ==",
    "license": "MIT",
    "engines": {
        "node": ">= 0.6"
    }
},
"node_modules/merge-descriptors": {
    "version": "1.0.3",
    "resolved": "https://registry.npmjs.org/merge-descriptors/-/merge-descriptors-1.0.3.tgz",
    "integrity": "sha512-gaNvAS7TZ897/rVaZ0nMtAyxNyi/pdbjbAwUpFQpN70GqnVfOiXpeUUMKRBMzXaSQ8DdTX4/0ms62r2K+hE6mQ==",
    "license": "MIT",
    "funding": {
        "url": "https://github.com/sponsors/sindresorhus"
    }
},
"node_modules/methods": {

```

```

"version": "1.1.2",
"resolved": "https://registry.npmjs.org/methods/-/methods-1.1.2.tgz",
"integrity": "sha512-icIAHeNqNm68zFtnZ0e+1L2yUldvzNoauKU4WBA3VvH/vPFieF7qfRlwUZU+DA9P9bPXIS90ulxoUoCH23sV2w==",
"license": "MIT",
"engines": {
  "node": ">= 0.6"
},
"node_modules/mime": {
  "version": "1.6.0",
  "resolved": "https://registry.npmjs.org/mime/-/mime-1.6.0.tgz",
  "integrity": "sha512-x0Vn8spl+wuJ1O6S7gnbaQg8Pxh4NNHb7KSINmEWKiPE4RKOpIvijn+NkmYmmRgP68mc70j2EbeTFRsrswaQeg==",
  "license": "MIT",
  "bin": {
    "mime": "cli.js"
  },
  "engines": {
    "node": ">=4"
  }
},
"node_modules/mime-db": {
  "version": "1.52.0",
  "resolved": "https://registry.npmjs.org/mime-db/-/mime-db-1.52.0.tgz",
  "integrity": "sha512-sPU4uV7dYlvtWJxwwxHD0PuihVNIE7TyAbQ5SWxDCB9mUYvOgroQOwYQQOKPJ8ClbE+1ETVIOoK1UC2nU3gYvg==",
  "license": "MIT",
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/mime-types": {
  "version": "2.1.35",
  "resolved": "https://registry.npmjs.org/mime-types/-/mime-types-2.1.35.tgz",
  "integrity": "sha512-ZDY+bPm5zTTF+YpCrAU9nK0UgICYPT0QtT1NZWFv4s++TNkcgVaT0g6+4R2ul4MjQjzysHB1zxuWL50hzaeXiw==",
  "license": "MIT",
}

```

```

"dependencies": {
  "mime-db": "1.52.0"
},
"engines": {
  "node": ">= 0.6"
}
},
"node_modules/minimatch": {
  "version": "3.1.2",
  "resolved": "https://registry.npmjs.org/minimatch/-/minimatch-3.1.2.tgz",
  "integrity": "sha512-J7p63hRiAjw1NDEww1W7i37+BylrOWO5XQQAzZ3VOcL0PNybwpfmV/N05zFAzwQ9USyEcX6t3UO+K5aqBQOIHw==",
  "license": "ISC",
  "dependencies": {
    "brace-expansion": "^1.1.7"
  },
  "engines": {
    "node": "*"
  }
},
"node_modules/minimist": {
  "version": "1.2.8",
  "resolved": "https://registry.npmjs.org/minimist/-/minimist-1.2.8.tgz",
  "integrity": "sha512-2yyAR8qBkN3YuheJanUpWC5U3bb5osDywNB8RzDVIDwDHbocAJveqqj1u8+SVD7jkWT4yvsHCpWqqWqAxb0zCA==",
  "license": "MIT",
  "optional": true,
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/mkdirp": {
  "version": "0.5.6",
  "resolved": "https://registry.npmjs.org/mkdirp/-/mkdirp-0.5.6.tgz",
  "integrity": "sha512-FP+p8RB8OWpF3YZBCrP5gtADmtXApB5AMLn+vdyA+PyxCjrCs00mjyUozssO33cwDeT3wNGdLxJ5M//YqtHAJw==",
  "license": "MIT",
  "optional": true,
  "dependencies": {

```

```

    "minimist": "^1.2.6"
},
"bin": {
    "mkdirp": "bin/cmd.js"
}
},
"node_modules/mkdirp-classic": {
    "version": "0.5.3",
    "resolved": "https://registry.npmjs.org/mkdirp-classic/-/mkdirp-classic-0.5.3.tgz",
    "integrity": "sha512-gKLcREMhtuZRwRAfqP3RFW+TK4JqApVBtOlftVgjuABpAtpxhPGaDcfvbhNvD0B8iD1oUr/txX35NjcaY6Ns/A==",
    "license": "MIT"
},
"node_modules/ms": {
    "version": "2.0.0",
    "resolved": "https://registry.npmjs.org/ms/-/ms-2.0.0.tgz",
    "integrity": "sha512-Tpp60P6IUJDTuOq/5Z8cdskzJujfwqfOTkrwlwj7IRISpnkJnT6SyJ4PCPnGMoFjC9ddhal5KVIYtAt97ix05A==",
    "license": "MIT"
},
"node_modules/negotiator": {
    "version": "0.6.3",
    "resolved": "https://registry.npmjs.org/negotiator/-/negotiator-0.6.3.tgz",
    "integrity": "sha512-+EUsqGPLsM+j/zdChZjsnX51g4XrHFOIXwfnCVPGIQk/k5giakcKsuxCObBRu6DSm9opw/O6sIWbJdghQM4bBg==",
    "license": "MIT",
    "engines": {
        "node": ">= 0.6"
    }
},
"node_modules/node-fetch": {
    "version": "2.7.0",
    "resolved": "https://registry.npmjs.org/node-fetch/-/node-fetch-2.7.0.tgz",
    "integrity": "sha512-c4FRfUm/dbcWZ7U+1Wq0AwCyFL+3nt2bEw05wfxSz+DWpWsigtgmSgYmy2dQdWyKC1694ELPqMs/YzUSNozLt8A==",
    "license": "MIT",
    "dependencies": {

```

```

    "whatwg-url": "^5.0.0"
},
"engines": {
    "node": "4.x || >=6.0.0"
},
"peerDependencies": {
    "encoding": "^0.1.0"
},
"peerDependenciesMeta": {
    "encoding": {
        "optional": true
    }
}
},
"node_modules/node-webpmux": {
    "version": "3.1.7",
    "resolved": "https://registry.npmjs.org/node-webpmux/-/node-webpmux-3.1.7.tgz",
    "integrity": "sha512-ySkL4IBCto86OyQ0bIAGzyIWSECcn5I0IM3bYEhe75T8Zxt/BFUMHa8ktUguR7zwXNdS/Hms31VfSsYKN1383g==",
    "license": "ISC"
},
"node_modules/node-windows": {
    "version": "1.0.0-beta.8",
    "resolved": "https://registry.npmjs.org/node-windows/-/node-windows-1.0.0-beta.8.tgz",
    "integrity": "sha512-uLekXnSeem3nW5esclD224Fd0U/1VtvE796JpSpOY+c73CsIz/Qn2WUHRJyPQJEMrNGAy/FMRFjjhh4z1alZTA==",
    "license": "MIT",
    "dependencies": {
        "xml": "1.0.1",
        "yargs": "^17.5.1"
    }
},
"node_modules/node-windows/node_modules/cliui": {
    "version": "8.0.1",
    "resolved": "https://registry.npmjs.org/cliui/-/cliui-8.0.1.tgz",
    "integrity": "sha512-BSeNnyus75C4//NQ9gQt1/csTXyo/8Sb+afLAkzAptFuMsod9HFokGNudZpi/oQV73hnVK+sR+5PVRMd+Dr7YQ==",

```

```

"license": "ISC",
"dependencies": {
  "string-width": "^4.2.0",
  "strip-ansi": "^6.0.1",
  "wrap-ansi": "^7.0.0"
},
"engines": {
  "node": ">=12"
}
},
"node_modules/node-windows/node_modules/wrap-ansi": {
  "version": "7.0.0",
  "resolved": "https://registry.npmjs.org/wrap-ansi/-/wrap-ansi-7.0.0.tgz",
  "integrity": "sha512-YVGlj2kamLSTxw6NsZjoBxfSwsn0ycdesmc4p+Q21c5zPuZ1pl+NfxVdxPtdHv mNVOQ6XSYG4AUyt/Fi7D16Q==",
  "license": "MIT",
  "dependencies": {
    "ansi-styles": "^4.0.0",
    "string-width": "^4.1.0",
    "strip-ansi": "^6.0.0"
},
"engines": {
  "node": ">=10"
},
"funding": {
  "url": "https://github.com/chalk/wrap-ansi?sponsor=1"
}
},
"node_modules/node-windows/node_modules/y18n": {
  "version": "5.0.8",
  "resolved": "https://registry.npmjs.org/y18n/-/y18n-5.0.8.tgz",
  "integrity": "sha512-0pfFzegeDWJHJIAmTLRP2DwHjdF5s7jo9tuztdQxAhINCdvS+3nGINqPd00Ap hqJR/0LhANUS6/+7SCb98YOfA==",
  "license": "ISC",
  "engines": {
    "node": ">=10"
}
},
"node_modules/node-windows/node_modules/yargs": {
  "version": "17.7.2",

```

```

"resolved": "https://registry.npmjs.org/yargs/-/yargs-17.7.2.tgz",
"integrity": "sha512-7dSzzRQ++CKnNI/krKnYRV7JKKPUXMEh61soaHKg9mrWEhzFWhFnxPxGI+69cD1Ou63C13NUPCnmlcrvqCuM6w==",
"license": "MIT",
"dependencies": {
  "cliui": "^8.0.1",
  "escalade": "^3.1.1",
  "get-caller-file": "^2.0.5",
  "require-directory": "^2.1.1",
  "string-width": "^4.2.3",
  "y18n": "^5.0.5",
  "yargs-parser": "^21.1.1"
},
"engines": {
  "node": ">=12"
},
"node_modules/node-windows/node_modules/yargs-parser": {
  "version": "21.1.1",
  "resolved": "https://registry.npmjs.org/yargs-parser/-/yargs-parser-21.1.1.tgz",
  "integrity": "sha512-tVpsJW7DdjecAiFpbIB1e3qxIQsE6NoPc5/eTdrbbIC4h0LVsWhnoa3g+m2HclBlujHzsxZ4VJVA+GUuc2/LBw==",
  "license": "ISC",
  "engines": {
    "node": ">=12"
  },
  "node_modules/normalize-path": {
    "version": "3.0.0",
    "resolved": "https://registry.npmjs.org/normalize-path/-/normalize-path-3.0.0.tgz",
    "integrity": "sha512-6eZs5Ls3WtCisHWp9S2GUy8dqkpGi4BVSz3GaqiE6ezub0512ESztXUwUB6C6IKbQkY2Pnb/mD4WYojCRwcwLA==",
    "license": "MIT",
    "optional": true,
    "engines": {
      "node": ">=0.10.0"
    }
  }
}

```

```
},
"node_modules/object-assign": {
  "version": "4.1.1",
  "resolved": "https://registry.npmjs.org/object-assign/-/object-assign-4.1.1.tgz",
  "integrity": "sha512-rJgTQnkUnH1sFw8yT6VSU3zD3sWmu6sZhlseY8VX+GRu3P6F7Fu+JNDoXfkIElbLJSnc3FUQHVe4cU5hj+BcUg==",
  "license": "MIT",
  "engines": {
    "node": ">=0.10.0"
  }
},
"node_modules/object-inspect": {
  "version": "1.13.4",
  "resolved": "https://registry.npmjs.org/object-inspect/-/object-inspect-1.13.4.tgz",
  "integrity": "sha512-W67iLI4J2EXEGTbfeHCffrjDfitvLANg0UIX3wFUUSTx92KXRFeGMHUVgSqE+wvhAbi4WqjGg9czysTV2Epbew==",
  "license": "MIT",
  "engines": {
    "node": ">= 0.4"
  },
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/on-finished": {
  "version": "2.4.1",
  "resolved": "https://registry.npmjs.org/on-finished/-/on-finished-2.4.1.tgz",
  "integrity": "sha512-oVlzk3ENAhCk2zdv7IJwd/QUD4z2RxRwpkcGY8psCVcCYZNq4wYnVWALHM+brtuJjePWiYF/CImuDr8Ch5+kg==",
  "license": "MIT",
  "dependencies": {
    "ee-first": "1.1.1"
  },
  "engines": {
    "node": ">= 0.8"
  }
},
```

```

"node_modules/once": {
  "version": "1.4.0",
  "resolved": "https://registry.npmjs.org/once/-/once-1.4.0.tgz",
  "integrity": "sha512-  
INaJgI+2Q5URQBkccEKHTQOPaXdUxnZZEIQTZY0MFUAuaEqe1E+Nyvgdz/a  
lyNi6Z9MzO5dv1H8n58/GELp3+w==",
  "license": "ISC",
  "dependencies": {
    "wrappy": "1"
  }
},
"node_modules/p-limit": {
  "version": "2.3.0",
  "resolved": "https://registry.npmjs.org/p-limit/-/p-limit-2.3.0.tgz",
  "integrity": "sha512-  
//88mFWSJx8IxCzwdAABTJL2MyWB12+eIY7MDL2SqLmAkeKU9qxRvWuSyTj  
m3FUmpBEMuFfckAlqEaVGUDxb6w==",
  "license": "MIT",
  "dependencies": {
    "p-try": "^2.0.0"
  },
  "engines": {
    "node": ">=6"
  },
  "funding": {
    "url": "https://github.com/sponsors/sindresorhus"
  }
},
"node_modules/p-locate": {
  "version": "4.1.0",
  "resolved": "https://registry.npmjs.org/p-locate/-/p-locate-4.1.0.tgz",
  "integrity": "sha512-  
R79ZZ/0wAxKGu3oYMIz8jy/kbhsNrS7SKZ7PxEHBgJ5+F2mtFW2fK2cOtBh1c  
HYkQsbzFV7I+EoRKe6Yt0oK7A==",
  "license": "MIT",
  "dependencies": {
    "p-limit": "^2.2.0"
  },
  "engines": {
    "node": ">=8"
  }
},

```

```

"node_modules/p-try": {
  "version": "2.2.0",
  "resolved": "https://registry.npmjs.org/p-try/-/p-try-2.2.0.tgz",
  "integrity": "sha512-R4nPAVTAU0B9D35/Gk3uJf/7XYbQcyohSKdvAxIRSNgFI4e71hVoGnBNQz9cWaXxO2I10KTC+3jMdvoKw6dQ==",
  "license": "MIT",
  "engines": {
    "node": ">=6"
  },
  "node_modules/parseurl": {
    "version": "1.3.3",
    "resolved": "https://registry.npmjs.org/parseurl/-/parseurl-1.3.3.tgz",
    "integrity": "sha512-CiyeOxF/JZyN5m0z9PfXw4SCBJ6Sygz1Dpl0wqjlhDEGGBP1GnsUVEL0p63hoG1fcj3fHynXi9NYO4nWOL+qQ==",
    "license": "MIT",
    "engines": {
      "node": ">= 0.8"
    },
    "node_modules/path-exists": {
      "version": "4.0.0",
      "resolved": "https://registry.npmjs.org/path-exists/-/path-exists-4.0.0.tgz",
      "integrity": "sha512-ak9Qy5Q7jYb2Wwcey5Fpvg2KoAc/ZlhLSLOSBmRmygPsGwkVVt0fZa0qrtMz+m6tJTAHfZQ8FnmB4MG4LWy7/w==",
      "license": "MIT",
      "engines": {
        "node": ">=8"
      },
      "node_modules/path-is-absolute": {
        "version": "1.0.1",
        "resolved": "https://registry.npmjs.org/path-is-absolute/-/path-is-absolute-1.0.1.tgz",
        "integrity": "sha512-AVbw3UJ2e9bq64vSaS9Am0fje1Pa8pbGqTTsmXfaliMpnr5DIDhfJOuLj9Sf95ZPVDAUerDfEk88MPmPe7UCQg==",
        "license": "MIT",
        "engines": {

```

```

    "node": ">=0.10.0"
  }
},
"node_modules/path-to-regexp": {
  "version": "0.1.12",
  "resolved": "https://registry.npmjs.org/path-to-regexp/-/path-to-regexp-0.1.12.tgz",
  "integrity": "sha512-RA1GjUVMnvYFxuqovrEqZoxxW5NUZqbwKtYz/Tt7nXerk0LbLbIQmrsgdeOxV5SFHf0UDggjS/bSeOZwt1pmEQ==",
  "license": "MIT"
},
"node_modules/pend": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/pend/-/pend-1.2.0.tgz",
  "integrity": "sha512-F3asv42UuXchdzt+xXqfW1OGIVBe+mxa2mql0pg5yAHZPvFmY3Y6drSf/GQ1A86WgWEN9Kzh/WrgKa6iGcHXLg==",
  "license": "MIT"
},
"node_modules/pngjs": {
  "version": "5.0.0",
  "resolved": "https://registry.npmjs.org/pngjs/-/pngjs-5.0.0.tgz",
  "integrity": "sha512-40QW5YalBNfQo5yRYmiw7Yz6TKKWr3h6970B2YE+3fQpsWcrbj1PzJgxeJ19DRQjhMbKPluMY8rFaXc8moolVw==",
  "license": "MIT",
  "engines": {
    "node": ">=10.13.0"
  }
},
"node_modules/process-nextick-args": {
  "version": "2.0.1",
  "resolved": "https://registry.npmjs.org/process-nextick-args/-/process-nextick-args-2.0.1.tgz",
  "integrity": "sha512-3ouUOpQhtgrbOa17J7+uxOTpITYWaGP7/AhoR3+A+/1e9skrzelGi/dXzEYyvbxubEF6Wn2ypscTKiKJFFn1ag==",
  "license": "MIT",
  "optional": true
},
"node_modules/progress": {

```

```

    "version": "2.0.3",
    "resolved": "https://registry.npmjs.org/progress/-/progress-2.0.3.tgz",
    "integrity": "sha512-7PiHtLII5LdnKIMw100I+8xJXR5gW2QwWYkT6iJva0bXitZKa/XMrSbdmg3r2Xnaidz9QuMd0VPAmrZIF9V9sA==",
    "license": "MIT",
    "engines": {
      "node": ">=0.4.0"
    },
    "node_modules/proxy-addr": {
      "version": "2.0.7",
      "resolved": "https://registry.npmjs.org/proxy-addr/-/proxy-addr-2.0.7.tgz",
      "integrity": "sha512-lIQsMLSUDUPT44jdrU/O37qlnifitDP+ZwrmmZcoSKyLKvtZxpyV0n2/bD/N4tBAAZ/gJEdZU7KMraoK1+XYAg==",
      "license": "MIT",
      "dependencies": {
        "forwarded": "0.2.0",
        "ipaddr.js": "1.9.1"
      },
      "engines": {
        "node": ">= 0.10"
      }
    },
    "node_modules/proxy-from-env": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/proxy-from-env/-/proxy-from-env-1.1.0.tgz",
      "integrity": "sha512-D+zkORCbA9f1tdWRK0RaCR3GPv50cMxcrz4X8k5LTSUD1Dkw47mKJEZQNunlRTkWwgtaUSo1RVFRIG9ZXiFYg==",
      "license": "MIT"
    },
    "node_modules/pump": {
      "version": "3.0.3",
      "resolved": "https://registry.npmjs.org/pump/-/pump-3.0.3.tgz",
      "integrity": "sha512-todwxLMY7/heScKmntwQG8CXVkJWUOdYxlvY2s0VWAAMh/nd8SoYiRaKjl7+iCs984f2P8zvrfWcDDYVb73NfA==",
      "license": "MIT",
      "dependencies": {

```

```

    "end-of-stream": "^1.1.0",
    "once": "^1.3.1"
  },
  "node_modules/puppeteer": {
    "version": "18.2.1",
    "resolved": "https://registry.npmjs.org/puppeteer/-/puppeteer-18.2.1.tgz",
    "integrity": "sha512-7+UhmYa7wxPh2oMRwA++k8UGVDxh3YdWFB52r9C3tM81T6BU7cuusUSxmz0GEYSOYUKk/YzlhkQ6+vc0gHbxQ==",
    "deprecated": "< 24.9.0 is no longer supported",
    "hasInstallScript": true,
    "license": "Apache-2.0",
    "dependencies": {
      "https-proxy-agent": "5.0.1",
      "progress": "2.0.3",
      "proxy-from-env": "1.1.0",
      "puppeteer-core": "18.2.1"
    },
    "engines": {
      "node": ">=14.1.0"
    }
  },
  "node_modules/puppeteer-core": {
    "version": "18.2.1",
    "resolved": "https://registry.npmjs.org/puppeteer-core/-/puppeteer-core-18.2.1.tgz",
    "integrity": "sha512-MRtTAZfQTluz3U2oU/X2VqVWPcR1+94nbA2V6ZrSZRVEwLqZ8eclZ551qGFQD/vD2PYqHJwWOW/fpC721uznVw==",
    "license": "Apache-2.0",
    "dependencies": {
      "cross-fetch": "3.1.5",
      "debug": "4.3.4",
      "devtools-protocol": "0.0.1045489",
      "extract-zip": "2.0.1",
      "https-proxy-agent": "5.0.1",
      "proxy-from-env": "1.1.0",
      "rimraf": "3.0.2",
      "tar-fs": "2.1.1",
      "unbzip2-stream": "1.4.3",
      "ws": "8.9.0"
    }
  }
}

```

```

},
"engines": {
  "node": ">=14.1.0"
}
},
"node_modules/puppeteer-core/node_modules/debug": {
  "version": "4.3.4",
  "resolved": "https://registry.npmjs.org/debug/-/debug-4.3.4.tgz",
  "integrity": "sha512-PRWFHuSU3eDtQJPvnNY7Jcket1j0t5OuOsFzPPzsekD52Zl8qUfFIP EiswXqlvHWGVHOgX+7G/vCNNhehwxfkQ==",
  "license": "MIT",
  "dependencies": {
    "ms": "2.1.2"
  },
  "engines": {
    "node": ">=6.0"
  },
  "peerDependenciesMeta": {
    "supports-color": {
      "optional": true
    }
  }
},
"node_modules/puppeteer-core/node_modules/ms": {
  "version": "2.1.2",
  "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.2.tgz",
  "integrity": "sha512-sGkPx+VjMtmA6MX27oA4FBFELFCZZ4S4XqeGOXCv68tT+jb3vk/RyaKWP0PTKyWtmLSM0b+adUTEvbs1PEaH2w==",
  "license": "MIT"
},
"node_modules/qrcode": {
  "version": "1.5.4",
  "resolved": "https://registry.npmjs.org/qrcode/-/qrcode-1.5.4.tgz",
  "integrity": "sha512-1ca71Zgiu6ORjHqFBDpnSMTR2ReToX4I1Au1VFLyVeBTFavzQnv5JxMFr3ukHVkpSrSA2MCk0INJSykjUfz7Zg==",
  "license": "MIT",
  "dependencies": {
    "dijkstrajs": "^1.0.1",
    "pngjs": "^5.0.0",
  }
}

```

```

  "yargs": "^15.3.1"
},
"bin": {
  "qrcode": "bin/qrcode"
},
"engines": {
  "node": ">=10.13.0"
}
},
"node_modules/qs": {
  "version": "6.13.0",
  "resolved": "https://registry.npmjs.org/qs/-/qs-6.13.0.tgz",
  "integrity": "sha512-+38ql9SOOr8tfZ4QmJNplMUxqjbe7LKvvZgWdExB0md+egZTtjLB67Gu0HRX3u/XOq7UU2Nx6nsjvS16Z9uwfp==",
  "license": "BSD-3-Clause",
  "dependencies": {
    "side-channel": "^1.0.6"
  },
  "engines": {
    "node": ">=0.6"
  },
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/range-parser": {
  "version": "1.2.1",
  "resolved": "https://registry.npmjs.org/range-parser/-/range-parser-1.2.1.tgz",
  "integrity": "sha512-Hrgsx+orqoygnmhFbKaHE6c296J+HTAQXoxEF6gNupROmmGJRoyzfG3ccAv eqCBrwr/2yxQ5BVd/GTI5agOwSg==",
  "license": "MIT",
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/raw-body": {
  "version": "2.5.2",
  "resolved": "https://registry.npmjs.org/raw-body/-/raw-body-2.5.2.tgz",

```

```

  "integrity": "sha512-8zGqypfENjClqGhgXToC8aB2r7YrBX+AQAfIPs/Mlk+BtPTztOvTS01NRW/3Eh60J+a48lt8qsCzirQ6loCVfA==",
  "license": "MIT",
  "dependencies": {
    "bytes": "3.1.2",
    "http-errors": "2.0.0",
    "iconv-lite": "0.4.24",
    "unpipe": "1.0.0"
  },
  "engines": {
    "node": ">= 0.8"
  },
  "node_modules/readable-stream": {
    "version": "3.6.2",
    "resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-3.6.2.tgz",
    "integrity": "sha512-9u/sniCrY3D5WdsERHzHE4G2YCXqoG5FTHUiCC4Slbr6XcLZBY05ya9EKjYe k9O5xOAwjGq+1JdGBAS7Q9ScoA==",
    "license": "MIT",
    "dependencies": {
      "inherits": "^2.0.3",
      "string_decoder": "^1.1.1",
      "util-deprecate": "^1.0.1"
    },
    "engines": {
      "node": ">= 6"
    }
  },
  "node_modules/readdir-glob": {
    "version": "1.1.3",
    "resolved": "https://registry.npmjs.org/readdir-glob/-/readdir-glob-1.1.3.tgz",
    "integrity": "sha512-v05I2k7xN8zXvPD9N+z/uhXPaj0sUFCe2rcWZIpBsqxfP7xFQ0tipAd/wjj1YxWyWtUS5IDJpOG82JKt2EAVA==",
    "license": "Apache-2.0",
    "optional": true,
    "dependencies": {
      "minimatch": "^5.1.0"
    }
  }
}

```

```

},
"node_modules/readdir-glob/node_modules/brace-expansion": {
  "version": "2.0.2",
  "resolved": "https://registry.npmjs.org/brace-expansion/-/brace-expansion-2.0.2.tgz",
  "integrity": "sha512-Jt0vHyM+jmUBqojB7E1NIYadt0vI0Qxjxd2TERw94wDz+E2LAm5vKMXXwg6ZZBTHPuUIDgQHKXvjGBdfcF1ZDQ==",
  "license": "MIT",
  "optional": true,
  "dependencies": {
    "balanced-match": "^1.0.0"
  },
  "node_modules/readdir-glob/node_modules/minimatch": {
    "version": "5.1.6",
    "resolved": "https://registry.npmjs.org/minimatch/-/minimatch-5.1.6.tgz",
    "integrity": "sha512-lKwV/1brpG6mBUFHtb7NUmtABCb2WZZmm2wNiOA5hAb8VdCS4B3dtMWyvcoViccwAW/COERjXLt0zP1zXUN26g==",
    "license": "ISC",
    "optional": true,
    "dependencies": {
      "brace-expansion": "^2.0.1"
    },
    "engines": {
      "node": ">=10"
    }
  },
  "node_modules/require-directory": {
    "version": "2.1.1",
    "resolved": "https://registry.npmjs.org/require-directory/-/require-directory-2.1.1.tgz",
    "integrity": "sha512-fGxEI7+wsG9xrvdjsrlmL22OMTTiHRwAMroiEeMgq8gzoLC/PQr7RsRDSTLUgbZAztF+TVlkHc6/4RlKrui+Q==",
    "license": "MIT",
    "engines": {
      "node": ">=0.10.0"
    }
  },
  "node_modules/require-main-filename": {

```

```

    "version": "2.0.0",
    "resolved": "https://registry.npmjs.org/require-main-filename/-/require-main-filename-2.0.0.tgz",
    "integrity": "sha512-NKN5kMDylKuldxYLSUfrbo5Tuzh4hd+2E8NPPX02mZtn1VuREQToYe/ZdlJy+J3uCpfaiGF05e7B8W0iXbQHmg==",
    "license": "ISC"
},
"node_modules/rimraf": {
    "version": "3.0.2",
    "resolved": "https://registry.npmjs.org/rimraf/-/rimraf-3.0.2.tgz",
    "integrity": "sha512-JZkJMZkAGFFPP2YqZXXPbMIMBgsxzE8ILs4IMIX/2o0L9UBw9O/Y3o6wFw/i9YLapcUJWwqbi3kdxIPdC62TIA==",
    "deprecated": "Rimraf versions prior to v4 are no longer supported",
    "license": "ISC",
    "dependencies": {
        "glob": "^7.1.3"
    },
    "bin": {
        "rimraf": "bin.js"
    },
    "funding": {
        "url": "https://github.com/sponsors/isaacs"
    }
},
"node_modules/safe-buffer": {
    "version": "5.2.1",
    "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.2.1.tgz",
    "integrity": "sha512-rp3So07KcdmmKbGvgaNxQSJr7bGVSvk5S9Eq1F+ppbRo70+YeaDxkw5Dd8NPN+GD6bjnYm2VuPuCXmpuYvmCXQ==",
    "funding": [
        {
            "type": "github",
            "url": "https://github.com/sponsors/feross"
        },
        {
            "type": "patreon",
            "url": "https://www.patreon.com/feross"
        },
        {

```

```

    "type": "consulting",
    "url": "https://feross.org/support"
  },
],
"license": "MIT"
},
"node_modules/safer-buffer": {
  "version": "2.1.2",
  "resolved": "https://registry.npmjs.org/safer-buffer/-/safer-buffer-2.1.2.tgz",
  "integrity": "sha512-YZo3K82SD7Riyi0E1EQPojLz7kpepnSQI9lyPbHHg1XXXevb5dJI7tpyN2ADxGcQbHG7vcyRHk0cbwqcQriUtg==",
  "license": "MIT"
},
"node_modules/send": {
  "version": "0.19.0",
  "resolved": "https://registry.npmjs.org/send/-/send-0.19.0.tgz",
  "integrity": "sha512-dW41u5VfLXu8SJh5bwRmyYUbAoSB3c9uQh6L8h/KtsFREPWpbX1lrljJo186Jc4nmci/sGUZ9a0a0J2zgfq2hw==",
  "license": "MIT",
  "dependencies": {
    "debug": "2.6.9",
    "depd": "2.0.0",
    "destroy": "1.2.0",
    "encodeurl": "~1.0.2",
    "escape-html": "~1.0.3",
    "etag": "~1.8.1",
    "fresh": "0.5.2",
    "http-errors": "2.0.0",
    "mime": "1.6.0",
    "ms": "2.1.3",
    "on-finished": "2.4.1",
    "range-parser": "~1.2.1",
    "statuses": "2.0.1"
  },
  "engines": {
    "node": ">= 0.8.0"
  }
},
"node_modules/send/node_modules/encodeurl": {
  "version": "1.0.2",

```

```

  "resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-1.0.2.tgz",
  "integrity": "sha512-TPJXq8JqFaVYm2CWmPvnP2Iyo4ZSM7/QKcSmuMLDObfpH5fi7RUGmd/rTDf+rut/saiDiQEeVTNgAmJEdAOx0w==",
  "license": "MIT",
  "engines": {
    "node": ">= 0.8"
  },
  "node_modules/send/node_modules/ms": {
    "version": "2.1.3",
    "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
    "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTIA==",
    "license": "MIT"
  },
  "node_modules/serve-static": {
    "version": "1.16.2",
    "resolved": "https://registry.npmjs.org/serve-static/-/serve-static-1.16.2.tgz",
    "integrity": "sha512-VqpJZKadQB/PEbEwvFdO43Ax5dFBZ2UECszz8bQ7pi7wt//PWe1P6MN7eCnjsatYtBT6EuiClbjSWP2WrIoTw==",
    "license": "MIT",
    "dependencies": {
      "encodeurl": "~2.0.0",
      "escape-html": "~1.0.3",
      "parseurl": "~1.3.3",
      "send": "0.19.0"
    },
    "engines": {
      "node": ">= 0.8.0"
    }
  },
  "node_modules/set-blocking": {
    "version": "2.0.0",
    "resolved": "https://registry.npmjs.org/set-blocking/-/set-blocking-2.0.0.tgz",
    "integrity": "sha512-KiKBS8AnWGEyLzofFfmvKwpdPzqiy16LvQfK3yv/fVH7Bj13/wl3JSR1J+rfgRE9q7xUJK4qvgS8raSOeLUehw==",
    "license": "ISC"
  }
}

```

```

"node_modules/setimmediate": {
  "version": "1.0.5",
  "resolved": "https://registry.npmjs.org/setimmediate/-/setimmediate-1.0.5.tgz",
  "integrity": "sha512-MATJdZp8sLqDI/68LfQmbP8zKPLQNV6BIZolgrscFDQ+RsvK/BxeDQOgyxKKoh0y/8h3BqVFnCqQ/gd+reiXA==",
  "license": "MIT",
  "optional": true
},
"node_modules/setprototypeof": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/setprototypeof/-/setprototypeof-1.2.0.tgz",
  "integrity": "sha512-E5LDX7Wrp85Kil5bhZv46j8jOeboKq5JMmYM3gVGdGH8xFpPWXUMsNrlODCrkoxMEeNi/XZIwuRvY4XNwYMJpw==",
  "license": "ISC"
},
"node_modules/side-channel": {
  "version": "1.1.0",
  "resolved": "https://registry.npmjs.org/side-channel/-/side-channel-1.1.0.tgz",
  "integrity": "sha512-ZX99e6tRweoUXqR+VBrslhda51Nh5MTQwou5tnUDgbtyM0dBgmhEDtWGP/xbKn6hqfPRHujUNwz5fy/wbbhnpw==",
  "license": "MIT",
  "dependencies": {
    "es-errors": "^1.3.0",
    "object-inspect": "^1.13.3",
    "side-channel-list": "^1.0.0",
    "side-channel-map": "^1.0.1",
    "side-channel-weakmap": "^1.0.2"
  },
  "engines": {
    "node": ">= 0.4"
  },
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/side-channel-list": {

```

```

"version": "1.0.0",
  "resolved": "https://registry.npmjs.org/side-channel-list/-/side-channel-list-1.0.0.tgz",
    "integrity": "sha512-FCLHtRD/gnpCiCHEiJLOwdmFP+wzCmDEkc9y7NsYxeF4u7Btsn1ZuwgwJGxImImHicJArLP4R0yX4c2KCrMrTA==",
    "license": "MIT",
    "dependencies": {
      "es-errors": "^1.3.0",
      "object-inspect": "^1.13.3"
    },
    "engines": {
      "node": ">= 0.4"
    },
    "funding": {
      "url": "https://github.com/sponsors/ljharb"
    }
  },
  "node_modules/side-channel-map": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/side-channel-map/-/side-channel-map-1.0.1.tgz",
      "integrity": "sha512-VCjCNfgMsby3tTdo02nbtM/ewra6jPHmpThenkTYh8pG9ucZ/1P8So4u4FGBek/BjpOVsDCMoLA/iuBKIFXRA==",
      "license": "MIT",
      "dependencies": {
        "call-bound": "^1.0.2",
        "es-errors": "^1.3.0",
        "get-intrinsic": "^1.2.5",
        "object-inspect": "^1.13.3"
      },
      "engines": {
        "node": ">= 0.4"
      },
      "funding": {
        "url": "https://github.com/sponsors/ljharb"
      }
    },
    "node_modules/side-channel-weakmap": {
      "version": "1.0.2",

```

```

  "resolved": "https://registry.npmjs.org/side-channel-weakmap/-/side-
channel-weakmap-1.0.2.tgz",
  "integrity": "sha512-
WPS/HvHQTYnHisLo9McqBHOJk2FkHO/tlpvldyrnem4aeQp4hai3gythswg6p01
oSoTI58rcpiFAjF2br2Ak2A==",
  "license": "MIT",
  "dependencies": {
    "call-bound": "^1.0.2",
    "es-errors": "^1.3.0",
    "get-intrinsic": "^1.2.5",
    "object-inspect": "^1.13.3",
    "side-channel-map": "^1.0.1"
  },
  "engines": {
    "node": ">= 0.4"
  },
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/statuses": {
  "version": "2.0.1",
  "resolved": "https://registry.npmjs.org/statuses/-/statuses-2.0.1.tgz",
  "integrity": "sha512-
RwNA9Z/7PrK06rYLIzFMIaF+I73iwpzsqRIFgbMLbTcLD6cOao82TaWefPXQvB
2fOC4AjuYSEndS7N/mTCbkdQ==",
  "license": "MIT",
  "engines": {
    "node": ">= 0.8"
  },
  "node_modules/string_decoder": {
    "version": "1.3.0",
    "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-
1.3.0.tgz",
    "integrity": "sha512-
hkRX8U1WjJFd8LsDJ2yQ/wWWxaopEsABU1XfkM8A+j0+85JAGppt16cr1Whg
6Klbb4okU6Mql6BOj+uup/wKeA==",
    "license": "MIT",
    "dependencies": {
      "safe-buffer": "~5.2.0"
    }
  }
}

```

```

},
"node_modules/string-width": {
  "version": "4.2.3",
  "resolved": "https://registry.npmjs.org/string-width/-/string-width-4.2.3.tgz",
  "integrity": "sha512-wKyQRQpjJ0slp62ErSZdGsjMJWsap5oRNihHhu6G7JVO/9jIB6UyevL+tXuOqrng8j/cxKTWyWUwvSTriiZz/g==",
  "license": "MIT",
  "dependencies": {
    "emoji-regex": "^8.0.0",
    "is-fullwidth-code-point": "^3.0.0",
    "strip-ansi": "^6.0.1"
  },
  "engines": {
    "node": ">=8"
  }
},
"node_modules/strip-ansi": {
  "version": "6.0.1",
  "resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-6.0.1.tgz",
  "integrity": "sha512-Y38VPSHcqqFrCpFnQ9vuSXmquuv5oXOKpGeT6aGrr3o3Gc9AlVa6JBfUSOCnbxGGZF+/0ool7KrPuUSztUdU5A==",
  "license": "MIT",
  "dependencies": {
    "ansi-regex": "^5.0.1"
  },
  "engines": {
    "node": ">=8"
  }
},
"node_modules/tar-fs": {
  "version": "2.1.1",
  "resolved": "https://registry.npmjs.org/tar-fs/-/tar-fs-2.1.1.tgz",
  "integrity": "sha512-V0r2Y9scmbDRLCNex/+hYzvp/zyYjvFbHPNgVTKfQvVrb6guiE/fxP+XbIDNR011utopbkex2nM4dHNv6GDsng==",
  "license": "MIT",
  "dependencies": {
    "chownr": "^1.1.1",
    "mkdirp-classic": "^0.5.2",
    "pump": "^3.0.0",
  }
}

```

```

    "tar-stream": "^2.1.4"
  },
},
"node_modules/tar-stream": {
  "version": "2.2.0",
  "resolved": "https://registry.npmjs.org/tar-stream/-/tar-stream-2.2.0.tgz",
  "integrity": "sha512-ujeqbceABgwMZxEJnk2HDY2DInUZ+9oEcb1KzTVfYHio0UE6dG71n60d8D2I4qNvleWrrXpmjpt7vZeF1LnMZQ==",
  "license": "MIT",
  "dependencies": {
    "bl": "^4.0.3",
    "end-of-stream": "^1.4.1",
    "fs-constants": "^1.0.0",
    "inherits": "^2.0.3",
    "readable-stream": "^3.1.1"
  },
  "engines": {
    "node": ">=6"
  }
},
"node_modules/through": {
  "version": "2.3.8",
  "resolved": "https://registry.npmjs.org/through/-/through-2.3.8.tgz",
  "integrity": "sha512-w89qg7PI8wAdvX60bMDP+bFoD5Dvhm9oLheFp5O4a2QF0cSBGsBX4qZmadPMvVqlLJBBCi+WqGGOAPvcDeNSVg==",
  "license": "MIT"
},
"node_modules/toidentifier": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/toidentifier/-/toidentifier-1.0.1.tgz",
  "integrity": "sha512-o5sSPKEkg/DIQNmH43V0/uerLrpzVedkUh8tGNvaeXpfpuwjKenlSox/2O/BTIZUtEe+JG7s5YhEz608PIAHRA==",
  "license": "MIT",
  "engines": {
    "node": ">=0.6"
  }
},
"node_modules/tr46": {
  "version": "0.0.3",

```

```

  "resolved": "https://registry.npmjs.org/tr46/-/tr46-0.0.3.tgz",
  "integrity": "sha512-  

N3WMsuqV66lT30CrXNbEjx4GEwlow3v6rr4mCcv6prnfwhS01rkgyFdjPNBYd9  

br7LpXV1+Emh01fHnq2Gdgrw==",
  "license": "MIT"
},
"node_modules/traverse": {
  "version": "0.3.9",
  "resolved": "https://registry.npmjs.org/traverse/-/traverse-0.3.9.tgz",
  "integrity": "sha512-  

iaWgk0hLP3SxGKDfnDJf8wTz4p2qlmnyihM5Hh/sGvQ3K37dPi/w8sRhdNIxYA  

1TwFwc5mDhIjq+O0RsvXBKdQ==",
  "license": "MIT/X11",
  "optional": true,
  "engines": {
    "node": "*"
  }
},
"node_modules/type-is": {
  "version": "1.6.18",
  "resolved": "https://registry.npmjs.org/type-is/-/type-is-1.6.18.tgz",
  "integrity": "sha512-  

TkRKr9sUTxEH8MdfuCSP7VizJyzRNMjj2J2do2Jr3Kym598JVdEksuzPQCnlFP  

W4ky9Q+iA+ma9BGm06XQBy8g==",
  "license": "MIT",
  "dependencies": {
    "media-typer": "0.3.0",
    "mime-types": "~2.1.24"
  },
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/unbzip2-stream": {
  "version": "1.4.3",
  "resolved": "https://registry.npmjs.org/unbzip2-stream/-/unbzip2-stream-1.4.3.tgz",
  "integrity": "sha512-  

mlExGW4w71ebDJviH16lQLtZS32VKqsSfk80GCfUlwT/4/hNRFsoscrF/c++9xin  

kMzECL1uL9DDwXqFWkruPg==",
  "license": "MIT",
  "dependencies": {

```

```

    "buffer": "^5.2.1",
    "through": "^2.3.8"
  },
  "node_modules/undici-types": {
    "version": "7.10.0",
    "resolved": "https://registry.npmjs.org/undici-types/-/undici-types-7.10.0.tgz",
    "integrity": "sha512-t5Fy/nfn+14LuOc2KNYg75vZqClpAiqscVvMygNnlSHBFpSXdJaYtXMcdNLpl/Qvc3P2cB3s6lOV51nqsFq4ag==",
    "license": "MIT",
    "optional": true
  },
  "node_modules/universalify": {
    "version": "2.0.1",
    "resolved": "https://registry.npmjs.org/universalify/-/universalify-2.0.1.tgz",
    "integrity": "sha512-gptHNQghINnc/vTGIk0SOFGFNXw7JVrlRUtConJRlvaw6DuX0wO5Jeko9sWrMBhh+PsYAZ7oXAiOnf/UKogyiw==",
    "license": "MIT",
    "optional": true,
    "engines": {
      "node": ">= 10.0.0"
    }
  },
  "node_modules/unpipe": {
    "version": "1.0.0",
    "resolved": "https://registry.npmjs.org/unpipe/-/unpipe-1.0.0.tgz",
    "integrity": "sha512-pjy2bYhSsufwWIKwPc+l3cN7+wuJIK6uz0YdJEQIQDbl6jo/YIPi4mb8agUkVC8BF7V8NuzeypNqRksA3hztKQ==",
    "license": "MIT",
    "engines": {
      "node": ">= 0.8"
    }
  },
  "node_modules/unzipper": {
    "version": "0.10.14",
    "resolved": "https://registry.npmjs.org/unzipper/-/unzipper-0.10.14.tgz",
    "integrity": "sha512-ti4wZj+0bQTiX2KmKWuwj7lhV+2n//uXEotUmGuQqrVZSEGFMbl68+c6JCQ8aAmUWYvtHEz2A8K6wXvueR/6g==",
  }
}

```

```

"license": "MIT",
"optional": true,
"dependencies": {
  "big-integer": "^1.6.17",
  "binary": "~0.3.0",
  "bluebird": "~3.4.1",
  "buffer-indexof-polyfill": "~1.0.0",
  "duplexer2": "~0.1.4",
  "fstream": "^1.0.12",
  "graceful-fs": "^4.2.2",
  "listenercount": "~1.0.1",
  "readable-stream": "~2.3.6",
  "setimmediate": "~1.0.4"
}
},
"node_modules/unzipper/node_modules/readable-stream": {
  "version": "2.3.8",
  "resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-2.3.8.tgz",
  "integrity": "sha512-8p0AUk4XODglewSi0l8Epjs+EVnWiK7NoDIEGU0HhE7+ZyY8D1IMY7odu5IRrFXGg71L15KG8QrPmum45RTtdA==",
  "license": "MIT",
  "optional": true,
  "dependencies": {
    "core-util-is": "~1.0.0",
    "inherits": "~2.0.3",
    "isarray": "~1.0.0",
    "process-nextick-args": "~2.0.0",
    "safe-buffer": "~5.1.1",
    "string_decoder": "~1.1.1",
    "util-deprecate": "~1.0.1"
  }
},
"node_modules/unzipper/node_modules/safe-buffer": {
  "version": "5.1.2",
  "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.2.tgz",
  "integrity": "sha512-Gd2UZBJDkXIY7GbJxfsE8/nvKkUEU1G38c1siN6QP6a9PT9MmHB8GnpSCSmMJSoF8LOIrt8ud/wPtojys4G6+g==",
  "license": "MIT",
  "optional": true
}

```

```

},
"node_modules/unzipper/node_modules/string_decoder": {
  "version": "1.1.1",
  "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-1.1.1.tgz",
  "integrity": "sha512-n/ShnvDi6FHbbVfviro+WojiFzv+s8MPMHBczVePfUpDJLwoLT0ht1I4YwBCbi8pJAvEEdnkHyPyTP/mzRfwg==",
  "license": "MIT",
  "optional": true,
  "dependencies": {
    "safe-buffer": "~5.1.0"
  }
},
"node_modules/util-deprecate": {
  "version": "1.0.2",
  "resolved": "https://registry.npmjs.org/util-deprecate/-/util-deprecate-1.0.2.tgz",
  "integrity": "sha512-EPD5q1uXyFxJpCrLnCc1nHnq3gOa6DZBocAil2TaSCA7VCJ1UJDMagCzlKXNsUYfD1daK//LTEQ8xilbrHtcw==",
  "license": "MIT"
},
"node_modules/utils-merge": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/utils-merge/-/utils-merge-1.0.1.tgz",
  "integrity": "sha512-pMZTvIkT1d+TFGvDOqodOclx0QWkkgi6Tdoa8gC8ffGAaqz9pzPTZWAAybbsHHoED/ztMtkv/VoTYyShUn81hA==",
  "license": "MIT",
  "engines": {
    "node": ">= 0.4.0"
  }
},
"node_modules/vary": {
  "version": "1.1.2",
  "resolved": "https://registry.npmjs.org/vary/-/vary-1.1.2.tgz",
  "integrity": "sha512-BNGbWLfd0eUPabhkXUVm0j8uuvREyTh5ovRa/dyow/BqAbZJyC+5fU+lzQOzmAKzYqYRAISoRhdQr3elZ/PXqg==",
  "license": "MIT",
  "engines": {

```

```

    "node": ">= 0.8"
  },
},
"node_modules/webidl-conversions": {
  "version": "3.0.1",
  "resolved": "https://registry.npmjs.org/webidl-conversions/-/webidl-conversions-3.0.1.tgz",
  "integrity": "sha512-2JAn3z8AR6rjK8Sm8orRC0h/bcl/DqL7tRPdGZ4I1CjdF+EaMLmYxBHyXuKL849eucPFhvBoxMsflfOb8kxaeQ==",
  "license": "BSD-2-Clause"
},
"node_modules/whatsapp-web.js": {
  "version": "1.32.0",
  "resolved": "https://registry.npmjs.org/whatsapp-web.js/-/whatsapp-web.js-1.32.0.tgz",
  "integrity": "sha512-uXBT2WtUxUUZpOlevy/x+9JIIU7MSnPniqZC0NJq7k5VTjkl1GYWJZIDJSSYTvnw3JOZMn1wSizHqb9g/q5eeew==",
  "license": "Apache-2.0",
  "dependencies": {
    "@pedroslopez/moduleraid": "^5.0.2",
    "fluent-ffmpeg": "2.1.3",
    "mime": "^3.0.0",
    "node-fetch": "^2.6.9",
    "node-webpmux": "3.1.7",
    "puppeteer": "^18.2.1"
  },
  "engines": {
    "node": ">=18.0.0"
  },
  "optionalDependencies": {
    "archiver": "^5.3.1",
    "fs-extra": "^10.1.0",
    "unzipper": "^0.10.11"
  }
},
"node_modules/whatsapp-web.js/node_modules/mime": {
  "version": "3.0.0",
  "resolved": "https://registry.npmjs.org/mime/-/mime-3.0.0.tgz",

```

```

  "integrity": "sha512-jSCU7/VB1lolWBZe14aEYHU/+1UMEHoaO7qxCOVJOw9GgH72VAWppxNcjU+x9a2k3GSIBXNKxXQFqRvvZ7vr3A==",
  "license": "MIT",
  "bin": {
    "mime": "cli.js"
  },
  "engines": {
    "node": ">=10.0.0"
  }
},
"node_modules/whatwg-url": {
  "version": "5.0.0",
  "resolved": "https://registry.npmjs.org/whatwg-url/-/whatwg-url-5.0.0.tgz",
  "integrity": "sha512-saE57nupxk6v3HY35+jzBwYa0rKSy0XR8JSxZPwgLr7ys0IBzhGviA1/TUGJLmSVqs8pb9AnvICXEuOHLprYTtTw==",
  "license": "MIT",
  "dependencies": {
    "tr46": "~0.0.3",
    "webidl-conversions": "^3.0.0"
  }
},
"node_modules/which": {
  "version": "1.3.1",
  "resolved": "https://registry.npmjs.org/which/-/which-1.3.1.tgz",
  "integrity": "sha512-HxDYWq1MTIQBj3nw0cqssHoTNU267KlrDuGZ1WYlxDStUtKUhOajmh112/TZmHxxUfuJqPXSOm7tDyas0OSIQ==",
  "license": "ISC",
  "dependencies": {
    "isexe": "^2.0.0"
  },
  "bin": {
    "which": "bin/which"
  }
},
"node_modules/which-module": {
  "version": "2.0.1",
  "resolved": "https://registry.npmjs.org/which-module/-/which-module-2.0.1.tgz",

```

```

    "integrity": "sha512-
iBdZ57RDvnOR9AGBhML2vFZf7h8vmBjhoaZqODJBFWHVtKkDmKuHai3cx5P
gVMrX5YDNp27AofYbAwctSS+vhQ==",
    "license": "ISC"
},
"node_modules/wrap-ansi": {
    "version": "6.2.0",
    "resolved": "https://registry.npmjs.org/wrap-ansi/-/wrap-ansi-6.2.0.tgz",
    "integrity": "sha512-
r6lPcBGxZXIlcymEu7InxDMhdW0KDxpLgoFLcguasxCaJ/SOIZwINatK9KY/tf+Z
rlywOKU0UDj3ATXUBfxJXA==",
    "license": "MIT",
    "dependencies": {
        "ansi-styles": "^4.0.0",
        "string-width": "^4.1.0",
        "strip-ansi": "^6.0.0"
    },
    "engines": {
        "node": ">=8"
    }
},
"node_modules/wrappy": {
    "version": "1.0.2",
    "resolved": "https://registry.npmjs.org/wrappy/-/wrappy-1.0.2.tgz",
    "integrity": "sha512-
I4Sp/DRseor9wL6EvV2+TuQn63dMkPjZ/sp9XkghTEbV9KIPS1xUsZ3u7/IQO4
wxTCFB4bgpQPRcR3QCvezPcQ==",
    "license": "ISC"
},
"node_modules/ws": {
    "version": "8.9.0",
    "resolved": "https://registry.npmjs.org/ws/-/ws-8.9.0.tgz",
    "integrity": "sha512-
Ja7nszREasGaYUYCI2k4ICKIRTt+y7XuqVoHR44Ypl49TtryqbqvDMn5eqfW7
e6HzTukDRIsXqzVHScqRcafG==",
    "license": "MIT",
    "engines": {
        "node": ">=10.0.0"
    },
    "peerDependencies": {
        "bufferutil": "^4.0.1",
        "utf-8-validate": "^5.0.2"
    }
}

```

```

},
"peerDependenciesMeta": {
  "bufferutil": {
    "optional": true
  },
  "utf-8-validate": {
    "optional": true
  }
},
"node_modules/xml": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/xml/-/xml-1.0.1.tgz",
  "integrity": "sha512-huCv9IH9Tcf95zuYCsQraZtWnJvBtLVE0QHMOs8bWyZAFZNDcYjsPq1nEx8jKA9y+Beo9v+7OBPRisQTjinQMw==",
  "license": "MIT"
},
"node_modules/y18n": {
  "version": "4.0.3",
  "resolved": "https://registry.npmjs.org/y18n/-/y18n-4.0.3.tgz",
  "integrity": "sha512-JKhqTOwSrqNA1NY5ISztJ1GrBiUodLMmlZuLiDaMRJ+itFd+ABVE8XBjOvlWL+rSqNDC74LCSFmlb/U4UZ4hJQ==",
  "license": "ISC"
},
"node_modules/yargs": {
  "version": "15.4.1",
  "resolved": "https://registry.npmjs.org/yargs/-/yargs-15.4.1.tgz",
  "integrity": "sha512-aePbxDmcYW++PaqBsJ+HYUFwCdv4LVvdnhBy78E57PIor8/OVvhMrADFFE Dh8DHDFRv/O9i3IPhsENjO7QX0+A==",
  "license": "MIT",
  "dependencies": {
    "cliui": "^6.0.0",
    "decamelize": "^1.2.0",
    "find-up": "^4.1.0",
    "get-caller-file": "^2.0.1",
    "require-directory": "^2.1.1",
    "require-main-filename": "^2.0.0",
    "set-blocking": "^2.0.0",
    "string-width": "^4.2.0",
  }
}

```

```

    "which-module": "^2.0.0",
    "y18n": "^4.0.0",
    "yargs-parser": "^18.1.2"
},
"engines": {
    "node": ">=8"
}
},
"node_modules/yargs-parser": {
    "version": "18.1.3",
    "resolved": "https://registry.npmjs.org/yargs-parser/-/yargs-parser-18.1.3.tgz",
    "integrity": "sha512-o50j0JeToy/4K6OZcaQmW6lyXXKhq7csREXcDwk2omFPJEwUNOVtJKvmDr9EI1fAJZUyZcRF7kxGBWmRXudrCQ==",
    "license": "ISC",
    "dependencies": {
        "camelcase": "^5.0.0",
        "decamelize": "^1.2.0"
    },
    "engines": {
        "node": ">=6"
    }
},
"node_modules/yauzl": {
    "version": "2.10.0",
    "resolved": "https://registry.npmjs.org/yauzl/-/yauzl-2.10.0.tgz",
    "integrity": "sha512-p4a9l6X6nu6IhoGmBqAcbJy1mlC4j27vEPZX9F4L4/vZT3Lyq1VkfHw/V/PUcB9Buo+DG3iHkT0x3Qya58zc3g==",
    "license": "MIT",
    "dependencies": {
        "buffer-crc32": "~0.2.3",
        "fd-slicer": "~1.1.0"
    }
},
"node_modules/zip-stream": {
    "version": "4.1.1",
    "resolved": "https://registry.npmjs.org/zip-stream/-/zip-stream-4.1.1.tgz",
    "integrity": "sha512-9qv4rlDiopXg4E69k+vMHjNN63YFMe9sZMrdlvKnCjICRWeCBswPPMPUfx+ipsAWq1LXHe70RcbaHdJJpS6hyQ==",

```

```

"license": "MIT",
"optional": true,
"dependencies": {
  "archiver-utils": "^3.0.4",
  "compress-commons": "^4.1.2",
  "readable-stream": "^3.6.0"
},
"engines": {
  "node": ">= 10"
}
},
"node_modules/zip-stream/node_modules/archiver-utils": {
  "version": "3.0.4",
  "resolved": "https://registry.npmjs.org/archiver-utils/-/archiver-utils-3.0.4.tgz",
  "integrity": "sha512-KVgf4XQVrTjhyWmx6cte4RxonPLR9onExufI1jhvw/MQ4BB6lsZD5gT8Lq+u/+pRkWna/6JoHpiQioaqFP5Rzw==",
  "license": "MIT",
  "optional": true,
  "dependencies": {
    "glob": "^7.2.3",
    "graceful-fs": "^4.2.0",
    "lazystream": "^1.0.0",
    "lodash.defaults": "^4.2.0",
    "lodash.difference": "^4.5.0",
    "lodash.flatten": "^4.4.0",
    "lodash.isplainobject": "^4.0.6",
    "lodash.union": "^4.6.0",
    "normalize-path": "^3.0.0",
    "readable-stream": "^3.6.0"
  },
  "engines": {
    "node": ">= 10"
  }
}
}
|   └── package.json
{
  "name": "whatsapp-local",
  "version": "1.0.0",

```

```

"main": "server.js",
"type": "module",
"scripts": {
  "start": "node server.js",
  "dev": "nodemon server.js"
},
"dependencies": {
  "cors": "^2.8.5",
  "dotenv": "^16.4.5",
  "express": "^4.19.2",
  "node-windows": "^1.0.0-beta.8",
  "qrcode": "^1.5.4",
  "whatsapp-web.js": "^1.26.0"
}
}

```

| └─ server.js

```

import 'dotenv/config';
import express from 'express';
import cors from 'cors';
import qrcode from 'qrcode';
import fs from 'fs';
import path from 'path';
import pkg from 'whatsapp-web.js';
const { Client, LocalAuth } = pkg;

const app = express();
app.use(cors());
app.use(express.json());

let ready = false;
let qrDataUrl = null;
let lastRawQR = null; // texto QR, por si se requiere exportar
const eventLog = [];
function logEvent(type, data) {
  const entry = { ts: new Date().toISOString(), type, data };
  eventLog.push(entry);
  if (eventLog.length > 100) eventLog.shift();
  console.log(`[${entry.ts}] ${type}`, data || '');
}

function renderStatusHTML() {
  if (ready) {
    return '<h1>WhatsApp está listo!</h1>';
  } else if (qrDataUrl) {
    return `<h1>Escanea este QR con WhatsApp</h1>`;
  }
}

```

```

    } else {
        return '<h1> ✎ Generando QR...</h1><script>setTimeout(() =>
location.reload(), 2000);</script>';
    }
}

const SESSION_PATH = process.env.WHATS_SESSION_PATH || './session';
const client = new Client({
    authStrategy: new LocalAuth({ dataPath: SESSION_PATH }),
    puppeteer: { headless: true, args: ['--no-sandbox','--disable-setuid-
sandbox'] }
});

client.on('qr', async (qr) => {
    ready = false;
    lastRawQR = qr;
    qrDataURL = await qrcode.toDataURL(qr);
    logEvent('qr', 'Nuevo QR generado');
});

client.on('loading_screen', (percent, msg) => logEvent('loading_screen',
`${percent}% ${msg}`));
client.on('authenticated', () => logEvent('authenticated'));
client.on('auth_failure', (m) => logEvent('auth_failure', m));
client.on('ready', () => { ready = true; qrDataURL = null;
logEvent('ready'); });
client.on('change_state', (s) => logEvent('change_state', s));
client.on('disconnected', (r) => {
    ready = false;
    logEvent('disconnected', r);
    // Auto re-init tras breve espera
    setTimeout(() => {
        logEvent('reinit_attempt');
        client.initialize();
    }, 5000);
});

client.initialize();

// Rutas API
// Raíz muestra mismo contenido que /status
app.get('/', (req, res) => {
    res.send(renderStatusHTML());
});

app.get('/status', (req, res) => {
    res.send(renderStatusHTML());
});

```

```

app.get('/status-json', (req, res) => {
  res.json({ ready, qr: ready ? null : qrDataUrl });
});

// Últimos eventos para depuración
app.get('/debug', (req, res) => {
  res.json({ ready, hasQR: !!qrDataUrl, events: eventLog.slice(-30) });
});

// Devuelve texto crudo del QR (opcional para generar con otra app)
app.get('/raw-qr', (req, res) => {
  if (lastRawQR && !ready) return res.type('text/plain').send(lastRawQR);
  res.status(404).send('No QR');
});

// Reinicia el cliente sin borrar sesión
app.post('/restart', async (req, res) => {
  try {
    await client.destroy();
    ready = false; qrDataUrl = null; lastRawQR = null;
    client.initialize();
    res.json({ ok: true, restarted: true });
  } catch (e) { res.status(500).json({ ok: false, error: e.message }); }
});

// Borra sesión y reinicia (para cuando QR no vincula)
app.post('/reset-session', async (req, res) => {
  try {
    await client.destroy();
    ready = false; qrDataUrl = null; lastRawQR = null;

    // Resolver ruta de sesión según configuración
    const resolvedSessionPath = path.isAbsolute(SESSION_PATH)
      ? SESSION_PATH
      : path.join(process.cwd(), SESSION_PATH);

    // Borrar carpeta de sesión completa
    if (fs.existsSync(resolvedSessionPath)) {
      fs.rmSync(resolvedSessionPath, { recursive: true, force: true });
      logEvent('session_deleted', resolvedSessionPath);
    }

    // LocalAuth guarda credenciales en subcarpeta "wwebjs_auth" dentro
    // de dataPath
    const authDir = path.join(resolvedSessionPath, 'wwebjs_auth');
    if (fs.existsSync(authDir)) {
      fs.rmSync(authDir, { recursive: true, force: true });
    }
  }
});

```

```

        logEvent('auth_deleted', authDir);
    }

    client.initialize();
    res.json({ ok: true, reset: true });
} catch (e) { res.status(500).json({ ok: false, error: e.message }); }
});

app.post('/send', async (req, res) => {
try {
    const { to, message } = req.body || {};
    if (!to || !message) return res.status(400).json({ error: 'to y
message son requeridos' });
    if (!ready) return res.status(503).json({ error: 'No listo, escanea
el QR en /status' });

    const jid = String(to).includes('@c.us') ? String(to) : `${to}@c.us`;
    const r = await client.sendMessage(jid, String(message));
    res.json({ ok: true, id: r?.id?.id || null });
} catch (e) {
    res.status(500).json({ error: e.message });
}
});

const PORT = process.env.PORT || 3001;
app.listen(PORT, () => console.log(`API WhatsApp local:
http://localhost:${PORT} (session dir: ${SESSION_PATH})`));

// Manejo global de errores no capturados
process.on('unhandledRejection', (r) =>
console.error('UnhandledRejection', r));
process.on('uncaughtException', (e) => {
    console.error('UncaughtException', e);
});

```

### | └ unistall-service.cjs

```

var Service = require('node-windows').Service;

var svc = new Service({
    name: 'WhatsApp Bridge CRM',
    script: 'C:\\xampp\\htdocs\\CRM_INT\\CRM\\bridge-node\\server.js'
});

svc.on('uninstall', function(){
    console.log('☒ Servicio WhatsApp Bridge desinstalado completamente');
});

```

```
console.log('☒ Desinstalando servicio WhatsApp Bridge...');  
svc.uninstall();
```

```
├─ config/  
│  └─ CorreoExpress.php
```

```
<?php  
use PHPMailer\PHPMailer\PHPMailer;  
use PHPMailer\PHPMailer\Exception;  
  
require_once __DIR__ . '/../LIB/phpmailer/PHPMailer.php';  
require_once __DIR__ . '/../LIB/phpmailer/SMTP.php';  
require_once __DIR__ . '/../LIB/phpmailer/Exception.php';  
  
/**  
 * Envía un e-mail con un código Express (6 dígitos) que expira en 5 min.  
 * @return bool true si se envió sin errores.  
 */  
function enviarCodigoExpress(string $destinatario, string $nombre, string  
$codigo): bool  
{  
    $mail = new PHPMailer(true);  
  
    try {  
        $mail->isSMTP();  
        $mail->Host = 'smtp-relay.brevo.com';  
        $mail->SMTPAuth = true;  
        $mail->Username = '91e0a3001@smtp-brevo.com'; // Este es tu  
correo Brevo  
        $mail->>Password = 'Apxa06dYz4NP1X9f'; // Tu clave SMTP de Brevo  
        $mail->SMTPSecure = 'tls';  
        $mail->Port = 587;  
        $mail->SMTPOptions = [  
            'ssl' => [  
                'verify_peer' => false,  
                'verify_peer_name' => false,  
                'allow_self_signed' => true  
            ]  
        ];  
  
        // IMPORTANTE: el remitente debe ser el mismo correo del SMTP o  
uno verificado en Brevo  
        $mail->setFrom('anuelmorera@gmail.com', 'Restaurante Bastos');  
  
        $mail->addAddress($destinatario, $nombre);  
        $mail->isHTML(true);  
        $mail->CharSet = 'UTF-8';  
        $mail->Subject = '🚀 Código Express (5 min)';
```

```

$mail->Body = "
<div style='font-family:Arial, sans-serif; font-size:16px'>
    <p>Hola <strong>$nombre</strong>,</p>
    <p>Tu código Express para redimir tu beneficio es:</p>
        <h2 style='letter-spacing:4px;
color:#f9c41f;'>$codigo</h2>
    <p>El código expira en <strong>5 minutos</strong>.</p>
    <br>
    <p style='font-size:12px; color:gray'>
        Si vos no solicitaste este código, ignorá este correo.
    </p>
</div>
";

$mail->send();
return true;
} catch (Exception $e) {
    error_log("Mailer Error (Express): " . $mail->ErrorInfo);
    return false;
}
}
}

```

```

|   |— CorreoHelper.php
<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require_once __DIR__ . '/../LIB/phpmailer/PHPMailer.php';
require_once __DIR__ . '/../LIB/phpmailer/SMTP.php';
require_once __DIR__ . '/../LIB/phpmailer/Exception.php';

function enviarCorreoCumple($destinatario, $nombre, $mensaje) {
    $mail = new PHPMailer(true);

    try {
        $mail->isSMTP();
        $mail->Host = 'smtp-relay.brevo.com';
        $mail->SMTPAuth = true;
        $mail->Username = '91e0a3001@smtp-brevo.com'; // Este es tu
correo Brevo
        $mail->Password = 'Apxa06dYz4NP1X9f'; // Tu clave SMTP de Brevo
        $mail->SMTPSecure = 'tls';
        $mail->Port = 587;
        $mail->SMTPOptions = [
            'ssl' => [
                'verify_peer' => false,

```

```

        'verify_peer_name' => false,
        'allow_self_signed' => true
    ]
];

        // IMPORTANTE: el remitente debe ser el mismo correo del SMTP o
        uno verificado en Brevo
        $mail->setFrom('anuelmorera@gmail.com', 'Restaurante Bastos');

        $mail->addAddress($destinatario, $nombre);
        $mail->isHTML(true);
        $mail->CharSet = 'UTF-8';
        $mail->Subject = '🎉 ¡Feliz Cumpleaños!';

        $mail->Body =
            <div style='font-family: Arial, sans-serif; font-size:
16px;'>
                <p>¡Hola $nombre! 🎉</p>
                <p>Todo el equipo del <strong>Restaurante Bastos</strong>
te desea un <strong>feliz cumpleaños</strong>.</p>
                <p>Hoy celebramos tu día especial y queremos consentirte
como te merecés. 🎉</p>
                <p><strong>¡Presentá este mensaje en el restaurante y
elegí entre una comida, una bebida o un postre gratis!</strong>
🍽️ 🍷 🎉</p>
                <br>
                <p style='font-size: 12px; color: gray;'>Promoción válida
únicamente hoy para nuestros clientes VIP. ¡Te esperamos con gusto!</p>
            </div>
        ";
    }

    $mail->send();
    return true;
} catch (Exception $e) {
    // Mostrar el error real para depurar
    echo "Mailer Error: " . $mail->ErrorInfo;
    return false;
}
}

```

| └ Database.php

```

<?php
// Archivo: config/Database.php
class Database
{
    private $host = "localhost";

```

```

private $db_name = "bd_bastos_crm";
private $username = "root";
private $password = "";
public $conn;

public function getConnection()
{
    $this->conn = null;
    try {
        $this->conn = new PDO(
            "mysql:host=" . $this->host . ";dbname=" . $this-
>db_name,
            $this->username,
            $this->password
        );
        $this->conn->exec("set names utf8");
        $this->conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
    } catch (PDOException $exception) {
        echo "Connection error: " . $exception->getMessage();
    }
    return $this->conn;
}
}

```

```

|_
|   └── controller/
|       └── AnalisisController.php

```

```

<?php
// controller/AnalisisController.php

// Establece el tipo de contenido de la respuesta como JSON
header('Content-Type: application/json');

// Incluye la configuración de la base de datos y los DAOs necesarios
require_once __DIR__ . '/../config/Database.php';
require_once __DIR__ . '/../model/cliente/ClienteDAO.php';
require_once __DIR__ . '/../model/compra/CompraDAO.php'; // Asegúrate de
tener este DAO y Mapper igual que los otros

try {
    // Obtiene la conexión a la base de datos
    $db = (new Database())->getConnection();
    if (!$db)
        throw new Exception("No se pudo conectar a la base de datos");

    // Determina la acción a ejecutar a partir de POST o GET
    $action = $_POST['action'] ?? $_GET['action'] ?? '';

```

```

switch ($action) {
    case 'clientesFrecuentes':
        // Obtiene todos los clientes
        $clienteDAO = new ClienteDAO($db);
        $clientes = $clienteDAO->readAll(); // array de objetos
ClienteDTO

        // Obtiene todas las compras
        $compraDAO = new CompraDAO($db);
        $compras = $compraDAO->readAll(); // array de objetos
CompraDTO

        // Inicializa la estructura para contar visitas y registrar
        la última visita por cliente
        $visitasPorCliente = [];
        foreach ($clientes as $cliente) {
            $visitasPorCliente[$cliente->id] = [
                'id' => $cliente->id,
                'nombre' => $cliente->nombre,
                'cedula' => $cliente->cedula,
                'telefono' => $cliente->telefono,
                'visitadas' => 0,
                'ultimaVisita' => null,
            ];
        }

        // Recorre todas las compras y actualiza el contador de
        visitas y la fecha de última visita
        foreach ($compras as $compra) {
            $cid = $compra->idCliente;
            if (isset($visitasPorCliente[$cid])) {
                $visitasPorCliente[$cid]['visitadas']++;
                // Actualiza la última visita si es más reciente
                if (
                    !$visitasPorCliente[$cid]['ultimaVisita'] ||
                    $compra->fechaCompra >
$visitasPorCliente[$cid]['ultimaVisita']
                ) {
                    $visitasPorCliente[$cid]['ultimaVisita'] =
$compra->fechaCompra;
                }
            }
        }

        // Filtra solo los clientes que han realizado al menos una
compra

```

```

        $ranking = array_filter($visitasPorCliente, fn($c) =>
$c['visitas'] > 0);
        // Ordena el ranking por cantidad de visitas descendente
        usort($ranking, fn($a, $b) => $b['visitas'] <=>
$a['visitas']));
        // Asigna la posición en el ranking
        $i = 1;
        foreach ($ranking as &$c)
            $c['posicion'] = $i++;

        // Obtiene el top 5 de clientes para el gráfico
        $top = array_slice($ranking, 0, 5);

        // Devuelve el resultado en formato JSON
        echo json_encode([
            'success' => true,
            'data' => [
                'ranking' => array_values($ranking), // todos los que
han comprado
                'top' => array_values($top)
            ]
        ]);
        break;

    case 'clientesMayorHistorial':
        // Obtiene clientes y compras
        $clienteDAO = new ClienteDAO($db);
        $compraDAO = new CompraDAO($db);

        $clientes = $clienteDAO->readAll();
        // Filtra solo clientes con totalHistorico > 0
        $ranking = array_filter($clientes, fn($c) => $c-
>totalHistorico > 0);
        // Ordena por totalHistorico descendente
        usort($ranking, fn($a, $b) => $b->totalHistorico <=>
$a-
>totalHistorico);

        // Obtiene todas las compras
        $compras = $compraDAO->readAll();

        // Indexa las compras por cliente (idCliente => array de
fechas)
        $comprasPorCliente = [];
        foreach ($compras as $compra) {
            $comprasPorCliente[$compra->idCliente][] = $compra-
>fechaCompra;
        }

```

```

        // Asigna posición y última compra a cada cliente en el
ranking
        $i = 1;
        foreach ($ranking as &$c) {
            $c->posicion = $i++;
            // Busca la fecha de la última compra
            if (!empty($comprasPorCliente[$c->id])) {
                $c->ultimaCompra = max($comprasPorCliente[$c->id]);
            } else {
                $c->ultimaCompra = null;
            }
        }
        unset($c); // Limpia la referencia

        // Obtiene el top 7 para el gráfico
        $top = array_slice($ranking, 0, 7);

        // Devuelve el resultado en formato JSON
        echo json_encode([
            'success' => true,
            'data' => [
                'ranking' => array_values($ranking),
                'top' => array_values($top)
            ]
        ]);
        break;
    case 'clientesInactivos':
        // Define el período de inactividad (en días). Para pruebas,
se usa un valor muy bajo (30 segundos)
        //$diasInactivo = 30; // <--- CAMBIA AQUÍ EL PERÍODO
        $diasInactivo = 0.0003; // 30 segundos PARA PRUEBA

        $clienteDAO = new ClienteDAO($db);
        $compraDAO = new CompraDAO($db);
        $clientes = $clienteDAO->readAll();
        $compras = $compraDAO->readAll();

        // Indexa las compras por cliente
        $comprasPorCliente = [];
        foreach ($compras as $compra) {
            $comprasPorCliente[$compra->idCliente][] = $compra-
>fechaCompra;
        }

        $inactivos = [];
        $hoy = new DateTime();

        // Recorre todos los clientes y determina si están inactivos

```

```

foreach ($clientes as $c) {
    // Busca la última compra (la más reciente)
    $ultimaCompra = null;
    if (!empty($comprasPorCliente[$c->id])) {
        $ultimaCompra = max($comprasPorCliente[$c->id]);
        $diasSinComprar = (new DateTime($ultimaCompra))->diff($hoy)->days;
    } else {
        // Si nunca ha comprado, se puede excluir o incluir
        // según la lógica deseada
        $diasSinComprar = null;
    }

    // Si el cliente supera el período de inactividad, se
    // agrega al listado
    if ($diasSinComprar !== null && $diasSinComprar >
$diasInactivo) {
        $inactivos[] = [
            'id' => $c->id,
            'nombre' => $c->nombre,
            'cedula' => $c->cedula,
            'telefono' => $c->telefono,
            'ultimaCompra' => $ultimaCompra,
            'diasSinComprar' => $diasSinComprar,
            'totalHistorico' => $c->totalHistorico,
        ];
    }
}

// Ordena los clientes inactivos por mayor tiempo sin comprar
usort($inactivos, fn($a, $b) => $b['diasSinComprar'] <=>
$a['diasSinComprar']);

// Obtiene el top 7 de inactivos
$top = array_slice($inactivos, 0, 7);

// Devuelve el resultado en formato JSON
echo json_encode([
    'success' => true,
    'diasInactivo' => $diasInactivo,
    'data' => [
        'ranking' => array_values($inactivos),
        'top' => array_values($top)
    ]
]);
break;

```

```

case 'residenciasFrecuentes':
    // Obtiene todos los clientes
    $clienteDAO = new ClienteDAO($db);
    $clientes = $clienteDAO->readAll();

    // Agrupa los clientes por residencia
    $porResidencia = [];
    foreach ($clientes as $c) {
        $res = trim($c->lugarResidencia) ?: '(SIN RESIDENCIA)';
        if (!isset($porResidencia[$res]))
            $porResidencia[$res] = [];
        $porResidencia[$res][] = [
            'id' => $c->id,
            'nombre' => $c->nombre,
            'cedula' => $c->cedula
        ];
    }

    // Construye el ranking de residencias más frecuentes
    $ranking = [];
    foreach ($porResidencia as $res => $clientesRes) {
        $ranking[] = [
            'residencia' => $res,
            'cantidad' => count($clientesRes),
            'clientes' => $clientesRes
        ];
    }

    // Ordena por cantidad de clientes descendente
    usort($ranking, fn($a, $b) => $b['cantidad'] <=>
    $a['cantidad']);

    // Obtiene el top 7 de residencias
    $top = array_slice($ranking, 0, 7);

    // Devuelve el resultado en formato JSON
    echo json_encode([
        'success' => true,
        'data' => [
            'ranking' => $ranking,
            'top' => $top
        ]
    ]);
    break;

case 'clientesAntiguos':
    // Obtiene todos los clientes
    $clienteDAO = new ClienteDAO($db);

```

```

$clientes = $clienteDAO->readAll();

    // Obtiene todas las compras
    $compraDAO = new CompraDAO($db);
    $compras = $compraDAO->readAll();

    $hoy = new DateTime();

    // Indexa las compras por cliente
    $comprasPorCliente = [];
    foreach ($compras as $c) {
        $cid = $c->idCliente;
        if (!isset($comprasPorCliente[$cid]))
            $comprasPorCliente[$cid] = [];
        $comprasPorCliente[$cid][] = $c;
    }

    $ranking = [];
    foreach ($clientes as $cli) {
        // Calcula la antigüedad del cliente en días desde su
        registro
        $fechaRegistro = new DateTime($cli->fechaRegistro);
        $antiguedadDias = $fechaRegistro->diff($hoy)->days;

        // Busca la última compra del cliente
        $ultCompra = null;
        if (isset($comprasPorCliente[$cli->id]) &&
count($comprasPorCliente[$cli->id])) {
            $ultCompraObj = array_reduce(
                $comprasPorCliente[$cli->id],
                function ($carry, $item) {
                    return (!$carry || $item->fechaCompra >
$carry->fechaCompra) ? $item : $carry;
                }
            );
            $ultCompra = $ultCompraObj->fechaCompra;
            $diasSinComprar = (new DateTime($ultCompra))-
>diff($hoy)->days;
        } else {
            $ultCompra = null;
            $diasSinComprar = $antiguedadDias;
        }

        // Agrega el cliente al ranking con los datos calculados
        $ranking[] = [
            'id' => $cli->id,
            'nombre' => $cli->nombre,
            'cedula' => $cli->cedula,
        ];
    }
}

```

```

        'antiguedadDias' => $antiguedadDias,
        'diasSinComprar' => $diasSinComprar,
        'totalHistorico' => $cli->totalHistorico,
        'ultimaCompra' => $ultCompra,
    ];
}

// Ordena el ranking por mayor antigüedad
usort($ranking, fn($a, $b) => $b['antiguedadDias'] <=>
$a['antiguedadDias']);

// Asigna posición y obtiene el top 7
$i = 1;
foreach ($ranking as &$c)
    $c['posicion'] = $i++;
$top = array_slice($ranking, 0, 7);

// Devuelve el resultado en formato JSON
echo json_encode([
    'success' => true,
    'data' => [
        'ranking' => array_values($ranking),
        'top' => array_values($top)
    ]
]);
break;

case 'ventasPorMes':
    // Obtiene todas las compras
    $compraDAO = new CompraDAO($db);
    $compras = $compraDAO->readAll();

    // Arreglo de nombres de meses en español
    $meses_es = [
        1 => 'Enero',
        2 => 'Febrero',
        3 => 'Marzo',
        4 => 'Abril',
        5 => 'Mayo',
        6 => 'Junio',
        7 => 'Julio',
        8 => 'Agosto',
        9 => 'Septiembre',
        10 => 'Octubre',
        11 => 'Noviembre',
        12 => 'Diciembre'
    ];

```

```

// Agrupa las compras por mes y año
$ventasPorMes = [];
foreach ($compras as $compra) {
    $fecha = new DateTime($compra->fechaCompra);
    $mes = (int) $fecha->format('m');
    $año = (int) $fecha->format('Y');
    $key = $año . '-' . $mes;
    if (!isset($ventasPorMes[$key])) {
        $ventasPorMes[$key] = [
            'mes' => $mes,
            'año' => $año,
            'total' => 0,
        ];
    }
    $ventasPorMes[$key]['total'] += $compra->total;
}

// Ordena los resultados por año y mes descendente
usort($ventasPorMes, function ($a, $b) {
    return ($b['año'] * 100 + $b['mes']) - ($a['año'] * 100 +
$a['mes']);
});

// Calcula la variación porcentual respecto al mes anterior
$ventasConVar = [];
for ($i = 0; $i < count($ventasPorMes); $i++) {
    $actual = $ventasPorMes[$i];
    $anterior = $ventasPorMes[$i + 1] ?? null;
    $variacion = null;
    if ($anterior && $anterior['total'] > 0) {
        $variacion = round(((($actual['total'] -
$anterior['total'])) / $anterior['total']) * 100, 1);
    }
    $ventasConVar[] = [
        'posicion' => $i + 1,
        'mes' => $meses_es[$actual['mes']],
        'año' => $actual['año'],
        'total' => $actual['total'],
        'variacion' => $variacion,
    ];
}

// Limita a los últimos 12 meses
$ventasConVar = array_slice($ventasConVar, 0, 12);

// Devuelve el resultado en formato JSON
echo json_encode([
    'success' => true,
]
);

```

```

        'data' => [
            'ventas' => $ventasConVar,
        ],
    ]);
break;

case 'ventasPorAnio':
// Obtiene todas las compras
$compraDAO = new CompraDAO($db);
$compras = $compraDAO->readAll();

// Agrupa las ventas por año
$ventasPorAnio = [];
foreach ($compras as $compra) {
    $fecha = new DateTime($compra->fechaCompra);
    $anio = (int) $fecha->format('Y');
    if (!isset($ventasPorAnio[$anio])) {
        $ventasPorAnio[$anio] = [
            'año' => $anio,
            'total' => 0,
        ];
    }
    $ventasPorAnio[$anio]['total'] += $compra->total;
}

// Ordena los años de mayor a menor
ksort($ventasPorAnio);

// Calcula la variación porcentual anual respecto al año
anterior
$ventasArray = array_values($ventasPorAnio);
$ventasConVar = [];
for ($i = 0; $i < count($ventasArray); $i++) {
    $actual = $ventasArray[$i];
    $anterior = $ventasArray[$i + 1] ?? null;
    $variacion = null;
    if ($anterior && $anterior['total'] > 0) {
        $variacion = round(((($actual['total'] -
$anterior['total']) / $anterior['total']) * 100, 1);
    }
    $ventasConVar[] = [
        ' posicion' => $i + 1,
        'año' => $actual['año'],
        'total' => $actual['total'],
        'variacion' => $variacion,
    ];
}

```

```

        // Limita a los últimos 10 años
        $ventasConVar = array_slice($ventasConVar, 0, 10);

        // Devuelve el resultado en formato JSON
        echo json_encode([
            'success' => true,
            'data' => [
                'ventas' => $ventasConVar,
            ],
        ]);
        break;

    default:
        // Acción no válida
        echo json_encode(['success' => false, 'message' => 'Acción no
válida']);
    }

} catch (Exception $e) {
    // Manejo de errores generales
    echo json_encode([
        'success' => false,
        'message' => 'Error del servidor: ' . $e->getMessage()
    ]);
}

```

## | └ BitacoraController.php

```

<?php
// Establece el tipo de contenido de la respuesta como JSON
header('Content-Type: application/json');

// Incluye la configuración de la base de datos y las clases necesarias
// para la bitácora
require_once __DIR__ . '/../config/Database.php';
require_once __DIR__ . '/../model/bitacora/BitacoraDAO.php';
require_once __DIR__ . '/../model/bitacora/BitacoraDTO.php';
require_once __DIR__ . '/../model/bitacora/BitacoraMapper.php';

// Controlador para gestionar la bitácora de sesiones de usuario
class BitacoraController
{
    // Registra la entrada de un usuario en la bitácora
    public function registrarEntrada($idUsuario)
    {
        $db = (new Database())->getConnection();
        $dao = new BitacoraDAO($db);

```

```

// Verifica si el usuario ya tiene una sesión activa
if ($dao->tieneSesionActiva($idUsuario)) {
    return false;
}

// Crea un nuevo registro de entrada
$dto = new BitacoraDTO();
$idUsuario = $idUsuario;
$dto->horaEntrada = date('H:i:s');
$dto->horaSalida = '00:00:00';
$dto->fecha = date('Y-m-d');

return $dao->create($dto);
}

// Registra la salida de un usuario en la bitácora
public function registrarSalida($idUsuario)
{
    $db = (new Database())->getConnection();
    $dao = new BitacoraDAO($db);
    $bitacoras = $dao->readByUsuario($idUsuario);
    $ok = false;

    // Busca el registro de sesión activa y actualiza la hora de
salida
    if (!empty($bitacoras)) {
        foreach ($bitacoras as $b) {
            if (empty($b->horaSalida) || $b->horaSalida ===
'00:00:00') {
                $b->horaSalida = date('H:i:s');
                $ok = $dao->update($b);
                break;
            }
        }
    }

    return $ok;
}
}

// ===== HTTP =====
// Si el archivo se ejecuta directamente vía HTTP (no CLI), procesa la
petición
if (php_sapi_name() !== 'cli' && basename(__FILE__) ===
basename($_SERVER['SCRIPT_FILENAME'])) {
    ob_start();
}

```

```

try {
    // Ejecuta limpieza automática de registros expirados
    BitacoraDAO::ejecutarLimpiezaAutomatica();

    // Conexión a la base de datos
    $db = (new Database())->getConnection();
    if (!$db) throw new Exception("No se pudo conectar a la base de
datos");

    $dao = new BitacoraDAO($db);
    $controller = new BitacoraController();
    // Determina la acción a ejecutar
    $action = $_POST['action'] ?? $_GET['action'] ?? '';
    $response = [];

    switch ($action) {
        case 'create':
            // No se permite crear registros directamente por esta
acción
            $response = ['success' => false, 'message' => "La acción
'create' directa no está permitida"];
            break;

        case 'read':
            // Lee un registro de bitácora por ID
            $id = $_POST['id'] ?? $_GET['id'] ?? '';
            if ($id === '') throw new Exception('ID requerido');
            $bitacora = $dao->read($id);
            $response = ['success' => $bitacora !== null, 'data' =>
$bitacora];
            break;

        case 'readByUsuario':
            // Lee todos los registros de bitácora de un usuario
            $idUsuario = $_POST['idUsuario'] ?? $_GET['idUsuario'] ??
 '';
            if ($idUsuario === '') throw new Exception('ID de usuario
requerido');
            $response = ['success' => true, 'data' => $dao-
>readByUsuario($idUsuario)];
            break;

        case 'readAll':
            // Lee todos los registros de bitácora
            $bitacoras = $dao->readAll();
            $data = [];

            // Formatea los datos para la respuesta

```

```

foreach ($bitacoras as $b) {
    $data[] = [
        'idUsuario' => $b->idUsuario,
        'nombreUsuario' => $b->nombreUsuario ?? '',
        'horaEntrada' => $b->horaEntrada,
        'horaSalida' => $b->horaSalida,
        'fecha' => $b->fecha
    ];
}

$response = ['success' => true, 'data' => $data];
break;

case 'update':
    // No se permite actualizar registros directamente por
esta acción
    $response = ['success' => false, 'message' => "La acción
'update' directa no está permitida"];
    break;

case 'registrarEntrada':
    // Registra la entrada de un usuario
    $idUsuario = $_POST['idUsuario'] ?? $_GET['idUsuario'] ??
    '';
    if ($idUsuario === '') throw new Exception('ID de usuario
requerido');
    $ok = $controller->registrarEntrada($idUsuario);
    $response = [
        'success' => $ok,
        'message' => $ok ? 'Entrada registrada' : 'Ya tienes
una sesión activa'
    ];
    break;

case 'registrarSalida':
    // Registra la salida de un usuario
    $idUsuario = $_POST['idUsuario'] ?? $_GET['idUsuario'] ??
    '';
    if ($idUsuario === '') throw new Exception('ID de usuario
requerido');
    $ok = $controller->registrarSalida($idUsuario);
    $response = [
        'success' => $ok,
        'message' => $ok ? 'Salida registrada' : 'No se pudo
registrar salida'
    ];
    break;

```

```

        case 'eliminarExpirados':
            // Elimina registros expirados de la bitácora
            $eliminados = $dao->eliminarExpirados();
            $response = [
                'success' => true,
                'message' => "$eliminados registros eliminados
correctamente.",
                'eliminados' => $eliminados
            ];
            break;

        default:
            // Acción no válida
            $response = ['success' => false, 'message' => "Acción no
válida en BitacoraController"];
    }

    // Devuelve la respuesta en formato JSON
    echo json_encode($response);

} catch (Exception $e) {
    // Manejo de errores generales
    echo json_encode([
        'success' => false,
        'message' => 'Error del servidor: ' . $e->getMessage()
    ]);
}

ob_end_flush();
}
?>

```

## | └─ ClienteController.php

```

<?php
// Archivo: controller/ClienteController.php

// Establece el tipo de contenido de la respuesta como JSON
header('Content-Type: application/json');

// Incluye los modelos y DAOs necesarios para la gestión de clientes y
códigos
require_once __DIR__ . '/../model/cliente/ClienteDAO.php';
require_once __DIR__ . '/../model/cliente/ClienteDTO.php';
require_once __DIR__ . '/../model/cliente/ClienteMapper.php';
require_once __DIR__ . '/../config/Database.php';

```

```

// Importa el modelo de código para la creación automática de códigos de
// barra
require_once __DIR__ . '/../model/codigo/CodigoDAO.php';
require_once __DIR__ . '/../model/codigo/CodigoDTO.php';

// Genera un código de barra único para un cliente usando su cédula y la
// fecha/hora actual
function generarCodigoBarra($cliente)
{
    // Ejemplo: cédula + fecha y hora actual para unicidad
    return $cliente->cedula . date('YmdHis');
}

// Valida una fecha de cumpleaños con reglas estrictas de formato y rango
function validarFechaCumpleEstricto(string $s): bool {
    // Formato exacto AAAA-MM-DD
    $dt = DateTime::createFromFormat('Y-m-d', $s);
    if (!$dt || $dt->format('Y-m-d') !== $s) return false;

    // Rango permitido de fechas
    $min = new DateTime('1900-01-01');
    $hoy = new DateTime('today');
    if ($dt < $min || $dt > $hoy) return false;

    // Edad permitida entre 0 y 120 años
    $edad = (int)$dt->diff($hoy)->y;
    return $edad >= 0 && $edad <= 120;
}

try {
    // Obtiene la conexión a la base de datos
    $db = (new Database())->getConnection();
    if (!$db)
        throw new Exception("No se pudo conectar a la base de datos");

    $dao = new ClienteDAO($db);
    // Determina la acción a ejecutar a partir de POST o GET
    $action = $_POST['action'] ?? $_GET['action'] ?? '';

    switch ($action) {
        case 'create':
            // Crea un nuevo cliente a partir de los datos recibidos
            $cliente = new ClienteDTO();
            $cliente->cedula = $_POST['cedula'] ?? '';
            $cliente->nombre = $_POST['nombre'] ?? '';
            $cliente->correo = $_POST['correo'] ?? '';
            $cliente->telefono = $_POST['telefono'] ?? '';
            $cliente->lugarResidencia = $_POST['lugarResidencia'] ?? '';

```

```

$cliente->fechaCumpleanos = $_POST['fechaCumpleanos'] ?? '';
$cliente->acumulado = $_POST['acumulado'] ?? 0;
$cliente->alergias = $_POST['alergias'] ?? '';
$cliente->gustosEspeciales = $_POST['gustosEspeciales'] ?? '';

';

        // Valida que el teléfono o correo no estén registrados para
        otro cliente
        if ($dao->existeTelefonoOCorreo($cliente->telefono, $cliente-
>correo)) {
            echo json_encode([
                'success' => false,
                'message' => 'El teléfono o correo ya están
registrados para otro cliente.'
            ]);
            exit;
        }
        // Valida que la cédula no esté registrada para otro cliente
        if ($dao->existeCedula($cliente->cedula, null)) {
            echo json_encode([
                'success' => false,
                'message' => 'La cédula ya está registrada.'
            ]);
            exit;
        }
        // Crea el cliente en la base de datos
        $result = $dao->create($cliente);

        if ($result) {
            // Obtiene el último ID insertado para el cliente
            $idCliente = $db->lastInsertId();

            // Crea automáticamente el código de barras asociado a
este cliente
            $codigoDAO = new CódigoDAO($db);
            $codigoDTO = new CódigoDTO();
            $codigoDTO->idCliente = $idCliente;
            $codigoDTO->codigoBarra = generarCódigoBarra($cliente);
            $codigoDTO->cantImpresiones = 0;
            $codigoDAO->create($codigoDTO);
        }

        // Devuelve el resultado de la operación
        echo json_encode([
            'success' => $result,
            'message' => $result ? 'Cliente creado exitosamente' :
'Error al crear cliente'
        ]);
    
```

```

        break;

    case 'readAll':
        // Devuelve todos los clientes registrados
        echo json_encode(['success' => true, 'data' => $dao->readAll()]);
        break;

    case 'read':
        // Devuelve los datos de un cliente por su ID
        $id = $_POST['id'] ?? $_GET['id'] ?? '';
        $cliente = $dao->read($id);
        echo json_encode(['success' => $cliente !== null, 'data' => $cliente]);
        break;

    case 'update':
        // Actualiza los datos de un cliente
        $cliente = new ClienteDTO();
        $cliente->id = $_POST['id'] ?? '';
        $cliente->cedula = $_POST['cedula'] ?? '';
        $cliente->nombre = $_POST['nombre'] ?? '';
        $cliente->correo = $_POST['correo'] ?? '';
        $cliente->telefono = $_POST['telefono'] ?? '';
        $cliente->lugarResidencia = $_POST['lugarResidencia'] ?? '';
        $cliente->fechaCumpleanos = $_POST['fechaCumpleanos'] ?? '';
        $cliente->acumulado = $_POST['acumulado'] ?? null;
        // Valida que el teléfono o correo no estén registrados para
        otro cliente
        if ($dao->existeTelefonoOCorreo($cliente->telefono, $cliente->correo, $cliente->id)) {
            echo json_encode([
                'success' => false,
                'message' => 'El teléfono o correo ya están
registrados para otro cliente.'
            ]);
            exit;
        }
        // Valida que la cédula no esté registrada para otro cliente
        if ($dao->existeCedula($cliente->cedula, $cliente->id)) {
            echo json_encode([
                'success' => false,
                'message' => 'La cédula ya está registrada.'
            ]);
            exit;
        }
        // Actualiza el cliente en la base de datos
        $result = $dao->update($cliente);

```

```

        echo json_encode(['success' => $result, 'message' => $result
? 'Cliente actualizado exitosamente' : 'Error al actualizar cliente']);
        break;

    case 'delete':
        // Elimina un cliente por su ID
        $id = $_POST['id'] ?? $_GET['id'] ?? '';
        $result = $dao->delete($id);
        echo json_encode(['success' => $result, 'message' => $result
? 'Cliente eliminado exitosamente' : 'Error al eliminar cliente']);
        break;

    case 'reassignCode':
        // Reasigna un código de barras a un cliente por motivo
        $idCliente = $_POST['idCliente'] ?? '';
        $motivo = $_POST['motivo'] ?? '';

        if (empty($idCliente) || empty($motivo)) {
            echo json_encode([
                'success' => false,
                'message' => 'Todos los campos son requeridos: ID
Cliente y Motivo'
            ]);
            break;
        }

        $codigoDAO = new CodigoDAO($db);
        $result = $codigoDAO->reassignCode($idCliente, $motivo);
        echo json_encode($result);
        break;

    case 'getHistorialReasignaciones':
        // Obtiene el historial de reasignaciones de códigos y todos
los clientes
        $codigoDAO = new CodigoDAO($db);

        // Obtener TODOS los códigos usando el procedimiento
existente
        $resultCódigos = $codigoDAO->readAll();

        // Obtener TODOS los clientes
        $resultClientes = $dao->readAll();

        if ($resultCódigos['success'] && $resultClientes) {
            echo json_encode([
                'success' => true,
                'data' => [
                    'códigos' => $resultCódigos['data'],

```

```

        'clientes' => $resultClientes
    ]
]);
} else {
    echo json_encode([
        'success' => false,
        'message' => 'Error al obtener datos'
    ]);
}
break;

default:
    // Acción no válida
    echo json_encode(['success' => false, 'message' => 'Acción no
válida']);
}
} catch (Exception $e) {
    // Manejo de errores generales
    echo json_encode([
        'success' => false,
        'message' => 'Error del servidor: ' . $e->getMessage()
    ]);
}

```

### | ┌── CódigoController.php

```

<?php
// controller/CódigoController.php

// Establece el tipo de contenido de la respuesta como JSON
header('Content-Type: application/json');

// Requerimientos de modelos y DAOs para códigos y clientes
require_once __DIR__ . '/../config/Database.php';
require_once __DIR__ . '/../model/código/CódigoDAO.php';
require_once __DIR__ . '/../model/código/CódigoDTO.php';
require_once __DIR__ . '/../model/código/CódigoMapper.php';
require_once __DIR__ . '/../model/cliente/ClienteDAO.php';
require_once __DIR__ . '/../model/cliente/ClienteDTO.php';
require_once __DIR__ . '/../model/cliente/ClienteMapper.php';

try {
    // Conexión a la base de datos
    $db = (new Database())->getConnection();
    if (!$db) {
        throw new Exception("No se pudo conectar a la base de datos");
    }

```

```

// Instancia DAO para códigos
$dao = new CodigoDAO($db);
// Determina la acción a ejecutar
$action = $_POST['action'] ?? $_GET['action'] ?? '';

switch ($action) {
    case 'create':
        // Crea un nuevo código de barras
        $codigo = new CodigoDTO();
        $codigo->idCliente = $_POST['idCliente'] ?? '';
        $codigo->codigoBarra = $_POST['codigoBarra'] ?? '';
        $codigo->cantImpresiones = $_POST['cantImpresiones'] ?? 0;

        $result = $dao->create($codigo);
        echo json_encode($result);
        break;

    case 'readAll':
        // Obtiene todos los códigos de barras y agrega info básica
        // del cliente
        $result = $dao->readAll();
        if ($result['success']) {
            $clienteDAO = new ClienteDAO($db);
            foreach ($result['data'] as &$codigo) {
                // Busca el cliente correspondiente al código
                $cliente = $clienteDAO->read($codigo->idCliente);
                if ($cliente && is_object($cliente)) {
                    // Si existe, agrega datos relevantes
                    $codigo->cedula = $cliente->cedula ?? '-';
                    $codigo->nombre = $cliente->nombre ?? '-';
                    $codigo->fechaRegistro = $cliente->fechaRegistro
                    ?? '-';
                } else {
                    // Si no existe, coloca valores por defecto
                    $codigo->cedula = '-';
                    $codigo->nombre = '-';
                    $codigo->fechaRegistro = '-';
                }
            }
            echo json_encode($result);
            break;
        }

    case 'read':
        // Obtiene un código de barras por su ID
        $id = $_POST['id'] ?? $_GET['id'] ?? '';
        $result = $dao->read($id);
        echo json_encode($result);
}

```

```

        break;

    case 'readByCliente':
        // Obtiene los códigos de barras asociados a un cliente
        $idCliente = $_POST['idCliente'] ?? $_GET['idCliente'] ?? '';
        $result = $dao->readByCliente($idCliente);
        echo json_encode($result);
        break;

    case 'update':
        // Actualiza los datos de un código de barras
        $codigo = new CodigoDTO();
        $codigo->id = $_POST['id'] ?? '';
        $codigo->codigoBarra = $_POST['codigoBarra'] ?? '';
        $codigo->cantImpresiones = $_POST['cantImpresiones'] ?? 0;

        $result = $dao->update($codigo);
        echo json_encode($result);
        break;

    case 'delete':
        // Elimina un código de barras por su ID
        $id = $_POST['id'] ?? $_GET['id'] ?? '';
        $result = $dao->delete($id);
        echo json_encode($result);
        break;

    case 'incrementarImpresiones':
        // Incrementa el contador de impresiones de un código de
        // barras
        if (isset($_POST['idCliente'])) {
            $idCliente = $_POST['idCliente'];

            $codigoExistente = $codigoDAO-
>obtenerPorIdCliente($idCliente);

            if ($codigoExistente) {
                $nuevaCantidad = $codigoExistente['cantImpresiones']
+ 1;
                $resultado = $codigoDAO-
>actualizarContadorImpresiones($idCliente, $nuevaCantidad);
            } else {
                $codigo = new Codigo();
                $codigo->setIdCliente($idCliente);
                $codigo->setCodigoBarra($idCliente);
                $codigo->setCantImpresiones(1);
                $resultado = $codigoDAO->create($codigo);
            }
        }

```

```

        if ($resultado) {
            echo json_encode(['success' => true, 'message' =>
'Contador actualizado']);
        } else {
            echo json_encode(['success' => false, 'message' =>
'Error al actualizar contador']);
        }
    } else {
        echo json_encode(['success' => false, 'message' => 'ID de
cliente requerido']);
    }
    break;

    default:
        // Acción no válida o no definida
        echo json_encode([
            'success' => false,
            'message' => 'Acción no válida o no definida en
CodigoController'
        ]);
    }
} catch (Exception $e) {
    // Manejo de errores generales
    echo json_encode([
        'success' => false,
        'message' => 'Error del servidor: ' . $e->getMessage()
    ]);
}

```

## |   └─ CompraController.php

```

<?php
// Archivo: controller/CompraController.php

// Establece el tipo de contenido de la respuesta como JSON
header('Content-Type: application/json');

// Requiere los modelos y DAOs necesarios para la gestión de compras
require_once __DIR__ . '/../model/compra/CompraDAO.php';
require_once __DIR__ . '/../model/compra/CompraDTO.php';
require_once __DIR__ . '/../model/compra/CompraMapper.php';
require_once __DIR__ . '/../config/Database.php';

try {
    // Conexión a la base de datos
    $db = (new Database())->getConnection();
    if (!$db)

```

```

        throw new Exception("No se pudo conectar a la base de datos");

        // Instancia el DAO para compras
        $dao = new CompraDAO($db);
        // Determina la acción a ejecutar
        $action = $_POST['action'] ?? $_GET['action'] ?? '';

        switch ($action) {
            case 'create':
                // Crea una nueva compra con los datos recibidos
                $compra = new CompraDTO();
                $compra->fechaCompra = $_POST['fechaCompra'] ?? date('Y-m-d');
                $compra->total = $_POST['total'] ?? 0;
                $compra->idCliente = $_POST['idCliente'] ?? null;
                $result = $dao->create($compra);
                echo json_encode(['success' => $result, 'message' => $result ? 'Compra registrada' : 'Error al registrar compra']);
                break;

            case 'read': // Buscar por IdCompra
                // Obtiene una compra por su ID
                $idCompra = $_POST['idCompra'] ?? $_GET['idCompra'] ?? '';
                $compra = $dao->read($idCompra);
                echo json_encode(['success' => $compra !== null, 'data' => $compra]);
                break;

            case 'delete':
                // Elimina una compra por su ID
                $idCompra = $_POST['idCompra'] ?? $_GET['idCompra'] ?? '';
                $result = $dao->delete($idCompra);
                echo json_encode(['success' => $result, 'message' => $result ? 'Compra eliminada' : 'Error al eliminar']);
                break;

            case 'readByCliente': // Historial de compras por cliente
                // Obtiene el historial de compras de un cliente
                $idCliente = $_POST['idCliente'] ?? $_GET['idCliente'] ?? '';
                $compras = $dao->readByCliente($idCliente);
                echo json_encode(['success' => true, 'data' => $compras]);
                break;

            case 'readAll':
                // Obtiene todas las compras registradas
                echo json_encode(['success' => true, 'data' => $dao->readAll()]);
                break;
        }
    }
}

```

```

        default:
            // Acción no válida
            echo json_encode(['success' => false, 'message' => 'Acción no
válida en CompraController']);
        }
    } catch (Exception $e) {
        // Manejo de errores generales
        echo json_encode([
            'success' => false,
            'message' => 'Error del servidor: ' . $e->getMessage()
        ]);
}

```

### |   └─ CumpleController.php

```

<?php
// Controlador para gestión de cumpleaños de clientes
header('Content-Type: application/json; charset=utf-8');
// Asegura respuestas JSON incluso ante errores inesperados
set_error_handler(function ($severity, $message, $file, $line) {
    http_response_code(500);
    echo json_encode([
        'success' => false,
        'message' => 'Error PHP: ' . $message,
        'where' => basename($file) . ':' . $line
    ]);
    exit;
});
set_exception_handler(function ($ex) {
    http_response_code(500);
    echo json_encode([
        'success' => false,
        'message' => 'Excepción: ' . $ex->getMessage()
    ]);
    exit;
});
// Requiere modelos y utilidades necesarios
require_once __DIR__ . '/../config/Database.php';
require_once __DIR__ . '/../model/cumple/CumpleDAO.php';
require_once __DIR__ . '/../model/cumple/CumpleDTO.php';
require_once __DIR__ . '/../model/cumple/CumpleMapper.php';
require_once __DIR__ . '/../config/CorreoHelper.php';
require_once __DIR__ . '/../LIB/phpmailer/WhatsAppService.php';

$action = $_POST['action'] ?? '';
try {

```

```

// Configuración de errores y logs
error_reporting(E_ALL);
ini_set('display_errors', 0); // No imprimir HTML de errores
ini_set('log_errors', 1);

// Conexión y DAO
$database = new Database();
$conn = $database->getConnection();
$dao = new CumpleDAO($conn);

switch ($action) {
    case 'readSemana':
        // Lee los cumpleaños de la semana (offset 0=actual,
        1=siguiente)
        $offset = isset($_REQUEST['offset']) ? (int)
$_REQUEST['offset'] : 0;
        if ($offset < 0)
            $offset = 0;
        if ($offset > 1)
            $offset = 1;

        $cumples = $dao->obtenerCumplesSemana($offset);

        header('Content-Type: application/json; charset=utf-8');
        echo json_encode(['success' => true, 'data' => $cumples]);
        break;
}

case 'cambiarEstado':
    // Cambia el estado de un registro de cumpleaños
    $id = $_POST['id'] ?? null;
    $nuevoEstado = $_POST['estado'] ?? null;

    if ($id === null || $nuevoEstado === null) {
        echo json_encode([
            'success' => false,
            'message' => 'Datos incompletos'
        ]);
        break;
    }

    $resultado = $dao->actualizarEstado($id, $nuevoEstado);
    echo json_encode([
        'success' => $resultado,
        'message' => $resultado ? 'Estado actualizado' : 'Error
al actualizar el estado'
    ]);
    break;
}

```

```

case 'marcarComoEnviado':
    // Marca un cumpleaños como enviado (estado LISTA)
    $id = $_POST['id'] ?? null;

    if ($id === null) {
        echo json_encode([
            'success' => false,
            'message' => 'ID no proporcionado'
        ]);
        break;
    }

    $resultado = $dao->actualizarEstado($id, 'LISTA');
    echo json_encode([
        'success' => $resultado,
        'message' => $resultado ? 'Estado cambiado a LISTA' :
'Error al actualizar'
    ]);
    break;

case 'enviarCorreoCumple':
    // Envía un correo de cumpleaños y registra el evento
    $correo = $_POST['correo'] ?? '';
    $nombre = $_POST['nombre'] ?? '';
    $mensaje = $_POST['mensaje'] ?? '';
    $idCliente = $_POST['idCliente'] ?? null;

    if (empty($correo) || empty($nombre) || empty($mensaje) ||
empty($idCliente)) {
        echo json_encode([
            'success' => false,
            'message' => 'Datos incompletos para enviar el
correo'
        ]);
        break;
    }

    $enviado = enviarCorreoCumple($correo, $nombre, $mensaje);

    if ($enviado) {
        // Calcula fecha de vencimiento (domingo de la semana)
        $hoy = new DateTime();
        $diaSemana = $hoy->format('w');
        $diasHastaDomingo = 7 - $diaSemana;
        $vence = clone $hoy;
        $vence->modify("+$diasHastaDomingo days");
        $venceStr = $vence->format('Y-m-d');
    }
}

```

```

        // Inserta en historial de cumpleaños
        $sql = "INSERT INTO cumple (IdCliente, FechaLlamada,
Vence, Vencido)
VALUES (:idCliente, CURDATE(), :vence, 'NO')";
$stmt = $conn->prepare($sql);
$stmt->bindParam(':idCliente', $idCliente);
$stmt->bindParam(':vence', $venceStr);
$stmt->execute();
}

echo json_encode([
    'success' => $enviado,
    'message' => $enviado ? 'Correo enviado correctamente' :
'Error al enviar el correo'
]);
break;

case 'enviarWhatsCumple':
    // Envía un WhatsApp de cumpleaños y registra el evento
    $telefono = $_POST['telefono'] ?? '';
    $nombre = $_POST['nombre'] ?? '';
    $mensaje = $_POST['mensaje'] ?? '';
    $idCliente = $_POST['idCliente'] ?? null;

    if (empty($telefono) || empty($nombre) || empty($idCliente))
{
    echo json_encode([
        'success' => false,
        'message' => 'Datos incompletos para enviar WhatsApp'
    ]);
    break;
}

// Normaliza el teléfono y agrega código país si falta
$telefonoLimpio = preg_replace('/\D+/', '', $telefono);
if (strlen($telefonoLimpio) < 8) {
    echo json_encode([
        'success' => false,
        'message' => 'Teléfono inválido para WhatsApp'
    ]);
    break;
}
$cc = getenv('DEFAULT_CC') ?: '506';
if (strpos($telefonoLimpio, $cc) !== 0) {
    if (strlen($telefonoLimpio) === 8) {
        $telefonoLimpio = $cc . $telefonoLimpio;
    }
}

```

```

}

try {
    $base = getenv('WHATS_BASE') ?: 'http://localhost:3001';
    $svc = new Whats ApiService($base);

    // Verifica que el servicio de WhatsApp esté listo
    $status = $svc->status();
    if (!($status['ready'] ?? false)) {
        echo json_encode([
            'success' => false,
            'message' => 'WhatsApp no está listo. Verifique que PM2 esté corriendo: pm2 status'
        ]);
        break;
    }

    // Usa mensaje por defecto si no viene desde el frontend
    $mensajeFinal = $mensaje ?: ("¡Hola $nombre! En Bastos sabemos que estás de cumpleaños. Visítanos para celebrarlo juntos y reclamar tu regalía 🎉🎁");
    $res = $svc->send($telefonoLimpio, $mensajeFinal);

    // Registra en historial igual que el flujo de correo
    $hoy = new DateTime();
    $diaSemana = $hoy->format('w');
    $diasHastaDomingo = 7 - $diaSemana;
    $vence = clone $hoy;
    $vence->modify("+{$diasHastaDomingo} days");
    $venceStr = $vence->format('Y-m-d');

    $sql = "INSERT INTO cumple (IdCliente, FechaLlamada, Vence, Vencido)
VALUES (:idCliente, CURDATE(), :vence, 'NO')";
$stmt = $conn->prepare($sql);
$stmt->bindParam(':idCliente', $idCliente);
$stmt->bindParam(':vence', $venceStr);
$stmt->execute();

echo json_encode([
    'success' => true,
    'message' => 'WhatsApp enviado correctamente',
    'result' => $res
]);
} catch (Exception $e) {
    echo json_encode([
        'success' => false,

```

```

'message' => 'Error enviando WhatsApp: ' . $e-
>getMessage()
    ]);
}
break;

case 'registrarLlamadaCumple':
    // Registra una llamada de cumpleaños y calcula vencimiento
    $idCliente = $_POST['idCliente'] ?? null;

    if (empty($idCliente)) {
        echo json_encode([
            'success' => false,
            'message' => 'ID del cliente no proporcionado'
        ]);
        break;
    }

    // Calcula fecha de vencimiento (domingo de la semana)
    $hoy = new DateTime();
    $diaSemana = $hoy->format('w');
    $diasHastaDomingo = 7 - $diaSemana;
    $vence = clone $hoy;
    $vence->modify("+$diasHastaDomingo days");
    $venceStr = $vence->format('Y-m-d');

    // Inserta en la tabla cumple
    $sql = "INSERT INTO cumple (IdCliente, FechaLlamada, Vence,
Vencido)
VALUES (:idCliente, CURDATE(), :vence, 'NO')";
$stmt = $conn->prepare($sql);
$stmt->bindParam(':idCliente', $idCliente);
$stmt->bindParam(':vence', $venceStr);
$stmt->execute();

// (Opcional) Actualiza el estado del cliente a 'LISTA'

echo json_encode([
    'success' => true,
    'message' => '¡Llamada registrada correctamente!'
]);
break;

case 'readHistorial':
    // Obtiene el historial de cumpleaños
    $historial = $dao->obtenerHistorial();
    echo json_encode($historial);
    break;

```

```

        default:
            // Acción no válida
            echo json_encode([
                'success' => false,
                'message' => 'Acción no válida'
            ]);
            break;
    }

} catch (Exception $e) {
    // Manejo de errores generales
    echo json_encode([
        'success' => false,
        'message' => 'Error general: ' . $e->getMessage()
    ]);
}

```

### | └─ ExpressController.php

```

<?php
// Controlador para generación y validación de códigos express
session_start(); // Inicia la sesión para almacenar códigos temporales
header('Content-Type: application/json'); // Respuestas en formato JSON

// Requiere configuración de base de datos y utilidades de correo
require_once __DIR__ . '/../config/Database.php';
require_once __DIR__ . '/../config/CorreoExpress.php';

// Determina la acción a ejecutar
$action = $_POST['action'] ?? $_GET['action'] ?? '';

switch ($action) {

    /* ----- 1. Generar y enviar ----- */
    case 'generarCodigoExpress':
        // Obtiene datos del cliente
        $idCliente = intval($_POST['idCliente'] ?? 0);
        $correo    = trim($_POST['correo'] ?? '');
        $nombre    = trim($_POST['nombre'] ?? '');

        // Valida que los datos sean correctos
        if (!$idCliente || !$correo || !$nombre) {
            echo json_encode(['success'=>false,'message'=>'Datos
incompletos.']);
            exit;
        }
}

```

```

// Genera un código de 6 dígitos aleatorio
$codigo = str_pad(strval(random_int(0, 999999)), 6, '0',
STR_PAD_LEFT);
$expires = time() + 300; // El código expira en 5 minutos

// Guarda el código y datos en la sesión
$_SESSION['express'] = [
    'idCliente' => $idCliente,
    'codigo'      => $codigo,
    'expires'     => $expires
];

// Envía el código por correo
$enviado = enviarCodigoExpress($correo, $nombre, $codigo);

echo json_encode(['success'=> $enviado]);
exit;

/* ----- 2. Validar ----- */
case 'validarCodigoExpress':
// Obtiene datos enviados por el usuario
$idCliente = intval($_POST['idCliente'] ?? 0);
$codigoCli = trim($_POST['codigo'] ?? '');

// Verifica que exista un código generado en sesión
if (!isset($_SESSION['express'])) {
    echo json_encode(['success'=>false, 'message'=>'No hay código
generado.']);
    exit;
}

$exp = $_SESSION['express'];

// Verifica si el código ha expirado
if ($exp['expires'] < time()) {
    unset($_SESSION['express']);
    echo json_encode(['success'=>false, 'message'=>'Código
expirado.']);
    exit;
}

// Valida que el código y el cliente coincidan
if ($exp['idCliente'] == $idCliente && $exp['codigo'] === $codigoCli)
{
    unset($_SESSION['express']);
    echo json_encode(['success'=>true]);
} else {

```

```

        echo json_encode(['success'=>false, 'message'=>'Código
incorrecto.']);
    }
    exit;

    default:
        // Acción no reconocida
        echo json_encode(['success'=>false, 'message'=>'Acción no
reconocida.']);
}

```

### | └─ MessageController.php

```

<?php
// Controlador de mensajes (DEPRECATED)
// Este archivo ya no se utiliza. La lógica fue movida a public/index.php

http_response_code(410); // Indica que el recurso ya no está disponible
header('Content-Type: application/json; charset=utf-8'); // Respuesta en
formato JSON
echo json_encode(['error'=>'MessageController eliminado. Usa
index.php']); // Mensaje de error descriptivo

```

### | └─ SessionController.php

```

<?php
// Controlador de sesión: gestiona login, cierre y verificación de sesión
de usuario.

session_start(); // Inicia o reanuda la sesión PHP
header('Content-Type: application/json'); // Responde siempre en formato
JSON

// Importa dependencias necesarias para bitácora y base de datos
require_once __DIR__ . '/../config/Database.php';
require_once __DIR__ . '/../model/bitacora/BitacoraDTO.php';
require_once __DIR__ . '/../model/bitacora/BitacoraDAO.php';
require_once __DIR__ . '/../model/bitacora/BitacoraMapper.php';
require_once __DIR__ . '/BitacoraController.php';

try {
    // Lee el cuerpo de la petición (JSON) y lo decodifica
    $input = file_get_contents('php://input');
    $data = json_decode($input, true);

    // Determina la acción a ejecutar (puede venir por JSON, POST o GET)
    $action = $data['action'] ?? $_POST['action'] ?? $_GET['action'] ??
';

```

```

$response = [];

switch ($action) {

    case 'login':
        // Iniciar sesión: requiere idUsuario
        $idUsuario = $data['idUsuario'] ?? null;

        if ($idUsuario) {
            // Marca la sesión como autenticada y guarda el idUsuario
            $_SESSION['authenticated'] = true;
            $_SESSION['idUsuario'] = $idUsuario;

            // Registra la entrada del usuario en la bitácora
            $bitacora = new BitacoraController();
            $bitacora->registrarEntrada($idUsuario);

            $response = ['success' => true, 'message' => 'Sesión iniciada'];
        } else {
            // Si falta el idUsuario, responde con error
            $response = ['success' => false, 'message' => 'ID de usuario requerido'];
        }
        break;

    case 'close_session':
        // Cerrar sesión: registra salida en bitácora si hay usuario
        if (isset($_SESSION['idUsuario'])) {
            $bitacora = new BitacoraController();
            $bitacora->registrarSalida($_SESSION['idUsuario']);
        }

        // Destruye la sesión y elimina cookies asociadas
        session_unset();
        session_destroy();

        if (ini_get("session.use_cookies")) {
            $params = session_get_cookie_params();
            setcookie(
                session_name(),
                '',
                time() - 42000,
                $params["path"],
                $params["domain"],
                $params["secure"],
                $params["httponly"]
            );
        }
}

```

```

    }

    $response = ['success' => true, 'message' => 'Sesión
cerrada'];
    break;

    case 'check_session':
        // Verifica si la sesión está activa y autenticada
        $active = isset($_SESSION['authenticated']) &&
$_SESSION['authenticated'] === true;
        $response = ['success' => true, 'active' => $active];
        break;

    default:
        // Acción no reconocida
        $response = ['success' => false, 'message' => 'Acción no
válida'];
    }

    // Devuelve la respuesta en formato JSON
    echo json_encode($response);
} catch (Exception $e) {
    // Manejo de errores generales: responde con mensaje de error
    echo json_encode([
        'success' => false,
        'message' => 'Error del servidor: ' . $e->getMessage()
    ]);
}
}

```

### |   └─ TareaController.php

```

<?php
// Controlador de tareas: gestiona la creación, consulta, actualización y
eliminación de tareas.

header('Content-Type: application/json'); // Responde siempre en formato
JSON

// Importa clases necesarias para el manejo de tareas y la base de datos
require_once __DIR__ . '/../model/tarea/TareaDAO.php';
require_once __DIR__ . '/../model/tarea/TareaDTO.php';
require_once __DIR__ . '/../model/tarea/TareaMapper.php';
require_once __DIR__ . '/../config/Database.php';

// Detecta si la petición POST llega como JSON puro y la decodifica
if ($_SERVER['REQUEST_METHOD'] === 'POST' && empty($_POST)) {
    $input = json_decode(file_get_contents("php://input"), true);
    if (json_last_error() === JSON_ERROR_NONE) {

```

```

        $_POST = $input; // Sobrescribe $_POST con los datos
        decodificados
    }
}

try {
    // Obtiene la conexión a la base de datos
    $db = (new Database())->getConnection();
    if (!$db) {
        throw new Exception("No se pudo conectar a la base de datos");
    }

    // Instancia el DAO de tareas
    $dao = new TareaDAO($db);
    // Determina la acción a ejecutar (puede venir por POST o GET)
    $action = $_POST['action'] ?? $_GET['action'] ?? '';

    switch ($action) {
        case 'create':
            // Crear una nueva tarea
            $descripcion = $_POST['descripcion'] ?? '';
            // Valida longitud máxima de la descripción
            if (mb_strlen($descripcion, 'UTF-8') > 220) {
                echo json_encode([
                    'success' => false,
                    'message' => 'La descripción supera el límite de 220
caracteres.'
                ]);
                exit;
            }

            // Crea el DTO de tarea y asigna valores
            $tarea = new TareaDTO();
            $tarea->descripcion = $descripcion;
            $tarea->estado = $_POST['estado'] ?? 'Pendiente';
            $tarea->fechaCreacion = date('Y-m-d H:i:s'); // Fecha actual

            // Inserta la tarea en la base de datos
            $result = $dao->create($tarea);

            echo json_encode([
                'success' => $result,
                'message' => $result ? 'Tarea creada' : 'Error al crear
tarea'
            ]);
            break;

        case 'readAll':
    }
}

```

```

        // Devuelve todas las tareas
        echo json_encode([
            'success' => true,
            'data' => $dao->readAll()
        ]);
        break;

    case 'read':
        // Devuelve una tarea específica por ID
        $id = $_POST['id'] ?? $_GET['id'] ?? '';
        $tarea = $dao->read($id);
        echo json_encode([
            'success' => $tarea !== null,
            'data' => $tarea
        ]);
        break;

    case 'update':
        // Actualiza el estado de una tarea
        $id = $_POST['id'] ?? '';
        $estado = $_POST['estado'] ?? '';
        $result = $dao->update($id, $estado);
        echo json_encode([
            'success' => $result,
            'message' => $result ? 'Tarea actualizada' : 'Error al
actualizar tarea'
        ]);
        break;

    case 'delete':
        // Elimina una tarea por ID
        $id = $_POST['id'] ?? $_GET['id'] ?? '';
        $result = $dao->delete($id);
        echo json_encode([
            'success' => $result,
            'message' => $result ? 'Tarea eliminada' : 'Error al
eliminar tarea'
        ]);
        break;

    default:
        // Acción no reconocida
        echo json_encode(['success' => false, 'message' => 'Acción no
válida']);
    }

} catch (Exception $e) {
    // Manejo de errores generales: responde con mensaje de error
    echo json_encode([

```

```

        'success' => false,
        'message' => 'Error del servidor: ' . $e->getMessage()
    ]);
}

```

| └ UsuarioController.php

```

<?php
// Controlador de usuarios: gestiona el CRUD, login, logout y
validaciones de usuarios del sistema.

header('Content-Type: application/json'); // Responde siempre en formato
JSON
date_default_timezone_set("America/Costa_Rica"); // Zona horaria para
fechas

// Importa dependencias necesarias para usuarios, bitácora y base de
datos
require_once __DIR__ . '/../config/Database.php';
require_once __DIR__ . '/../model/usuario/UsuarioDAO.php';
require_once __DIR__ . '/../model/usuario/UsuarioDTO.php';
require_once __DIR__ . '/../model/usuario/UsuarioMapper.php';
require_once __DIR__ . '/BitacoraController.php';

try {
    // Obtiene la conexión a la base de datos
    $db = (new Database())->getConnection();
    if (!$db) throw new Exception("No se pudo conectar a la base de
datos");

    // Instancia el DAO de usuarios
    $dao = new UsuarioDAO($db);
    // Determina la acción a ejecutar (puede venir por POST o GET)
    $action = $_POST['action'] ?? $_GET['action'] ?? '';

    switch ($action) {
        // Crear nuevo usuario
        case 'create':
            $usuario = trim($_POST['usuario'] ?? '');
            $contrasena = trim($_POST['contrasena'] ?? '');
            $rol = trim($_POST['rol'] ?? '');

            // Valida que los campos obligatorios estén presentes
            if ($usuario === '' || $contrasena === '' || $rol === '') {
                echo json_encode(['success' => false, 'message' =>
'Faltan campos obligatorios.']);
                return;
            }

```

```

        // Valida la fortaleza de la contraseña
        if (!preg_match('/^(?=.*[a-zA-Z])(?=.*\d).{6,}$/',
$contrasena)) {
            echo json_encode(['success' => false, 'message' => 'La
contraseña debe tener mínimo 6 caracteres, al menos 1 letra y 1
número.']);
            return;
        }

        // Verifica que el usuario no exista ya
        foreach ($dao->readAll() as $u) {
            if (strtolower($u->usuario) === strtolower($usuario)) {
                echo json_encode(['success' => false, 'message' =>
'El usuario ya existe.']);
                return;
            }
        }

        // Crea el DTO y asigna los valores
        $dto = new UsuarioDTO();
        $dto->usuario = $usuario;
        $dto->contrasena = password_hash($contrasena,
PASSWORD_DEFAULT);
        $dto->rol = $rol;

        // Guarda el usuario en la base de datos
        $ok = $dao->create($dto);
        echo json_encode([
            'success' => $ok === true,
            'message' => $ok ? 'Usuario guardado correctamente.' :
$ok
        ]);
        return;

        // Iniciar sesión
        case 'login':
            session_start(); // Inicia o reanuda la sesión
            $usuarioInput = trim($_POST['usuario'] ?? '');
            $contrasenaInput = $_POST['contrasena'] ?? '';

            // Valida que usuario y contraseña estén presentes
            if ($usuarioInput === '' || $contrasenaInput === '') {
                echo json_encode(['success' => false, 'message' =>
'Usuario y contraseña son requeridos']);
                return;
            }
    
```

```

    // Busca el usuario en la base de datos
    $usuarioEncontrado = null;
    foreach ($dao->readAll(false) as $u) {
        if ($u->usuario === $usuarioInput) {
            $usuarioEncontrado = $u;
            break;
        }
    }

    // Si no existe, responde con error
    if (!$usuarioEncontrado) {
        echo json_encode(['success' => false, 'message' =>
'Usuario no encontrado']);
        return;
    }

    // Verifica la contraseña
    if (!password_verify($contrasenaInput, $usuarioEncontrado-
>contrasena)) {
        echo json_encode(['success' => false, 'message' =>
'Contraseña incorrecta']);
        return;
    }

    // Guarda datos de sesión y registra entrada en bitácora
    $_SESSION['authenticated'] = true;
    $_SESSION['usuario'] = $usuarioEncontrado->usuario;
    $_SESSION['rol'] = $usuarioEncontrado->rol;
    $_SESSION['idUsuario'] = $usuarioEncontrado->id;
    $_SESSION['login_time'] = time();
    (new BitacoraController())-
>registrarEntrada($usuarioEncontrado->id);

    echo json_encode([
        'success' => true,
        'message' => 'Login exitoso',
        'usuario' => $usuarioEncontrado->usuario,
        'rol' => $usuarioEncontrado->rol,
        'redirect' => '/CRM_INT/CRM/index.php?view=dashboard'
    ]);
    return;

    // Leer todos los usuarios (sin contraseñas)
case 'readAll':
    $usuarios = $dao->readAll(true); // true: oculta contraseñas
    echo json_encode(['success' => true, 'data' => $usuarios]);
    return;

```

```

// Leer un usuario por ID
case 'read':
    $id = $_POST['id'] ?? $_GET['id'] ?? '';
    if ($id === '') {
        echo json_encode(['success' => false, 'message' => 'ID de
usuario requerido.']);
        return;
    }

    $usuario = $dao->read($id);
    if ($usuario) unset($usuario->contraseña); // Oculta la
contraseña
    echo json_encode(['success' => $usuario !== null, 'data' =>
$usuario]);
    return;

// Actualizar un usuario
case 'update':
    $id = trim($_POST['id'] ?? '');
    $usuario = trim($_POST['usuario'] ?? '');
    $contraseña = trim($_POST['contraseña'] ?? '');
    $rol = trim($_POST['rol'] ?? '');

    // Valida campos obligatorios
    if ($id === '' || $usuario === '' || $rol === '') {
        echo json_encode(['success' => false, 'message' =>
'Faltan campos obligatorios.']);
        return;
    }

    // Crea el DTO y asigna valores
    $dto = new UsuarioDTO();
    $dto->id = $id;
    $dto->usuario = $usuario;
    $dto->rol = $rol;
    // Si se proporciona nueva contraseña, la hashea; si no,
mantiene la anterior
    $dto->contraseña = $contraseña !== ''
        ? password_hash($contraseña, PASSWORD_DEFAULT)
        : ($dao->read($id, false)->contraseña ?? '');

    // Actualiza el usuario en la base de datos
    $ok = $dao->update($dto);
    echo json_encode([
        'success' => $ok,
        'message' => $ok ? 'Usuario actualizado correctamente.' :
'Error al actualizar usuario.'
    ]);

```

```

        return;

    // Eliminar usuario
    case 'delete':
        $id = $_POST['id'] ?? $_GET['id'] ?? '';
        if ($id === '') {
            echo json_encode(['success' => false, 'message' => 'ID de
usuario requerido.']);
            return;
        }

        // Elimina el usuario de la base de datos
        $ok = $dao->delete($id);
        echo json_encode([
            'success' => $ok,
            'message' => $ok ? 'Usuario eliminado correctamente.' :
'Error al eliminar usuario.'
        ]);
        return;

    // Cerrar sesión y registrar salida
    case 'logout':
        session_start();
        // Si hay usuario autenticado, registra salida en bitácora
        if (isset($_SESSION['idUsuario'])) {
            (new BitacoraController())-
>registrarSalida($_SESSION['idUsuario']);
        }
        session_destroy();

        // Si es una petición Beacon (al cerrar pestaña), no
redirecciona
        if (!isset($_SERVER['HTTP_SEC_FETCH_MODE']) ||

$_SERVER['HTTP_SEC_FETCH_MODE'] !== 'no-cors') {
            header('Location: /CRM_INT/CRM/index.php');
        }

        echo json_encode([
            'success' => true,
            'message' => 'Sesión cerrada correctamente.'
        ]);
        return;

    // Acción no válida
    default:
        // Acción no reconocida
        echo json_encode(['success' => false, 'message' => 'Acción no
válida']);

```

```

        return;

    }

} catch (Exception $e) {
    // Manejo de errores generales: responde con mensaje de error
    echo json_encode([
        'success' => false,
        'message' => 'Error del servidor: ' . $e->getMessage()
    ]);
}

```

```

|
|   └─ LIB/
|       └─ phpmailer/
|           └─ Exception.php

```

```

<?php

/**
 * PHPMailer Exception class.
 * PHP Version 5.5.
 *
 * @see      https://github.com/PHPMailer/PHPMailer/ The PHPMailer
GitHub project
 *
 * @author  Marcus Bointon (Synchro/coolbru)
<phpmailer@synchromedia.co.uk>
 * @author  Jim Jagielski (jimjag) <jimjag@gmail.com>
 * @author  Andy Prevost (codeworxtech)
<codeworxtech@users.sourceforge.net>
 * @author  Brent R. Matzelle (original founder)
 * @copyright 2012 - 2020 Marcus Bointon
 * @copyright 2010 - 2012 Jim Jagielski
 * @copyright 2004 - 2009 Andy Prevost
 * @license  https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html GNU
Lesser General Public License
 * @note This program is distributed in the hope that it will be
useful - WITHOUT
 * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
 * FITNESS FOR A PARTICULAR PURPOSE.
 */

namespace PHPMailer\PHPMailer;

/**
 * PHPMailer exception handler.
 *
```

```

 * @author Marcus Bointon <phpmailer@synchromedia.co.uk>
 */
class Exception extends \Exception
{
    /**
     * Prettify error message output.
     *
     * @return string
     */
    public function errorMessage()
    {
        return '<strong>' . htmlspecialchars($this->getMessage(),
ENT_COMPAT | ENT_HTML401) . "</strong><br />\n";
    }
}

```

```

|   └─ PHPMailer.php
<?php

/**
 * PHPMailer - PHP email creation and transport class.
 * PHP Version 5.5.
 *
 * @see https://github.com/PHPMailer/PHPMailer/ The PHPMailer GitHub
project
 *
 * @author Marcus Bointon (Synchro/coolbru)
<phpmailer@synchromedia.co.uk>
 * @author Jim Jagielski (jimjag) <jimjag@gmail.com>
 * @author Andy Prevost (codeworxtech)
<codeworxtech@users.sourceforge.net>
 * @author Brent R. Matzelle (original founder)
 * @copyright 2012 - 2020 Marcus Bointon
 * @copyright 2010 - 2012 Jim Jagielski
 * @copyright 2004 - 2009 Andy Prevost
 * @license https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html GNU
Lesser General Public License
 * @note This program is distributed in the hope that it will be useful -
WITHOUT
 * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
or
 * FITNESS FOR A PARTICULAR PURPOSE.
*/

```

```

namespace PHPMailer\PHPMailer;

/**
 * PHPMailer - PHP email creation and transport class.
 *
 * @author Marcus Bointon (Synchro/coolbru)
 <phpmailer@synchromedia.co.uk>
 * @author Jim Jagielski (jimjag) <jimjag@gmail.com>
 * @author Andy Prevost (codeworxtech)
 <codeworxtech@users.sourceforge.net>
 * @author Brent R. Matzelle (original founder)
 */
class PHPMailer
{
    const CHARSET_ASCII = 'us-ascii';
    const CHARSET_ISO88591 = 'iso-8859-1';
    const CHARSET_UTF8 = 'utf-8';

    const CONTENT_TYPE_PLAINTEXT = 'text/plain';
    const CONTENT_TYPE_TEXT_CALENDAR = 'text/calendar';
    const CONTENT_TYPE_TEXT_HTML = 'text/html';
    const CONTENT_TYPE_MULTIPART_ALTERNATIVE = 'multipart/alternative';
    const CONTENT_TYPE_MULTIPART_MIXED = 'multipart/mixed';
    const CONTENT_TYPE_MULTIPART RELATED = 'multipart/related';

    const ENCODING_7BIT = '7bit';
    const ENCODING_8BIT = '8bit';
    const ENCODING_BASE64 = 'base64';
    const ENCODING_BINARY = 'binary';
    const ENCODING_QUOTED_PRINTABLE = 'quoted-printable';

    const ENCRYPTION_STARTTLS = 'tls';
    const ENCRYPTION_SMTPS = 'ssl';

    const ICAL_METHOD_REQUEST = 'REQUEST';
    const ICAL_METHOD_PUBLISH = 'PUBLISH';
    const ICAL_METHOD_REPLY = 'REPLY';
    const ICAL_METHOD_ADD = 'ADD';
    const ICAL_METHOD_CANCEL = 'CANCEL';
    const ICAL_METHOD_REFRESH = 'REFRESH';
}

```

```

const ICAL_METHOD_COUNTER = 'COUNTER';
const ICAL_METHOD_DECLINECOUNTER = 'DECLINECOUNTER';

/*
 * Email priority.
 * Options: null (default), 1 = High, 3 = Normal, 5 = low.
 * When null, the header is not set at all.
 *
 * @var int|null
 */
public $Priority;

/*
 * The character set of the message.
 *
 * @var string
 */
public $CharSet = self::CHARSET_ISO88591;

/*
 * The MIME Content-type of the message.
 *
 * @var string
 */
public $ContentType = self::CONTENT_TYPE_PLAINTEXT;

/*
 * The message encoding.
 * Options: "8bit", "7bit", "binary", "base64", and "quoted-printable".
 *
 * @var string
 */
public $Encoding = self::ENCODING_8BIT;

/*
 * Holds the most recent mailer error message.
 *
 * @var string
 */
public $ErrorInfo = "";

/*

```

```

* The From email address for the message.
*
* @var string
*/
public $From = "";

/** 
 * The From name of the message.
*
* @var string
*/
public $FromName = "";

/** 
 * The envelope sender of the message.
* This will usually be turned into a Return-Path header by the receiver,
* and is the address that bounces will be sent to.
* If not empty, will be passed via '-f' to sendmail or as the 'MAIL FROM' value
over SMTP.
*
* @var string
*/
public $Sender = "";

/** 
 * The Subject of the message.
*
* @var string
*/
public $Subject = "";

/** 
 * An HTML or plain text message body.
* If HTML then call isHTML(true).
*
* @var string
*/
public $Body = "";

/** 
 * The plain-text message body.
* This body can be read by mail clients that do not have HTML email

```

```

* capability such as mutt & Eudora.
* Clients that can read HTML will view the normal Body.
*
* @var string
*/
public $AltBody = "";

/**
 * An iCal message part body.
 * Only supported in simple alt or alt_inline message types
 * To generate iCal event structures, use classes like EasyPeasyICS or
iCalcreator.
*
* @see https://kigkonsult.se/iCalcreator/
*
* @var string
*/
public $Ical = "";

/**
 * Value-array of "method" in Contenttype header "text/calendar"
*
* @var string[]
*/
protected static $IcalMethods = [
    self::ICAL_METHOD_REQUEST,
    self::ICAL_METHOD_PUBLISH,
    self::ICAL_METHOD_REPLY,
    self::ICAL_METHOD_ADD,
    self::ICAL_METHOD_CANCEL,
    self::ICAL_METHOD_REFRESH,
    self::ICAL_METHOD_COUNTER,
    self::ICAL_METHOD_DECLINECOUNTER,
];

/**
 * The complete compiled MIME message body.
*
* @var string
*/
protected $MIMEBody = "";

```

```

/***
 * The complete compiled MIME message headers.
 *
 * @var string
 */
protected $MIMEHeader = "";

/***
 * Extra headers that createHeader() doesn't fold in.
 *
 * @var string
 */
protected $mailHeader = "";

/***
 * Word-wrap the message body to this number of chars.
 * Set to 0 to not wrap. A useful value here is 78, for RFC2822 section 2.1.1
 * compliance.
 *
 * @see static::STD_LINE_LENGTH
 *
 * @var int
 */
public $WordWrap = 0;

/***
 * Which method to use to send mail.
 * Options: "mail", "sendmail", or "smtp".
 *
 * @var string
 */
public $Mailer = 'mail';

/***
 * The path to the sendmail program.
 *
 * @var string
 */
public $Sendmail = '/usr/sbin/sendmail';

/***
 * Whether mail() uses a fully sendmail-compatible MTA.

```

```

* One which supports sendmail's "-oi -f" options.
*
* @var bool
*/
public $UseSendmailOptions = true;

/**
 * The email address that a reading confirmation should be sent to, also known
as read receipt.
*
* @var string
*/
public $ConfirmReadingTo = "";

/**
 * The hostname to use in the Message-ID header and as default HELO string.
 * If empty, PHPMailer attempts to find one with, in order,
 * $_SERVER['SERVER_NAME'], gethostname(), php_uname('n'), or the
value
 * 'localhost.localdomain'.
*
* @see PHPMailer::$Hello
*
* @var string
*/
public $Hostname = "";

/**
 * An ID to be used in the Message-ID header.
 * If empty, a unique id will be generated.
 * You can set your own, but it must be in the format "<id@domain>",
* as defined in RFC5322 section 3.6.4 or it will be ignored.
*
* @see https://www.rfc-editor.org/rfc/rfc5322#section-3.6.4
*
* @var string
*/
public $MessageID = "";

/**
 * The message Date to be used in the Date header.
 * If empty, the current date will be added.

```

```

/*
 * @var string
 */
public $MessageDate = "";

/**
 * SMTP hosts.
 * Either a single hostname or multiple semicolon-delimited hostnames.
 * You can also specify a different port
 * for each host by using this format: [hostname:port]
 * (e.g. "smtp1.example.com:25;smtp2.example.com").
 * You can also specify encryption type, for example:
 * (e.g. "tls://smtp1.example.com:587;ssl://smtp2.example.com:465").
 * Hosts will be tried in order.
 *
 * @var string
 */
public $Host = 'localhost';

/**
 * The default SMTP server port.
 *
 * @var int
 */
public $Port = 25;

/**
 * The SMTP HELO/EHLO name used for the SMTP connection.
 * Default is $Hostname. If $Hostname is empty, PHPMailer attempts to find
 * one with the same method described above for $Hostname.
 *
 * @see PHPMailer::$Hostname
 *
 * @var string
 */
public $Hello = "";

/**
 * What kind of encryption to use on the SMTP connection.
 * Options:      ",      static::ENCRYPTION_STARTTLS,      or
static::ENCRYPTION_SMTPS.
 *

```

```

* @var string
*/
public $SMTPSecure = "";

/** 
 * Whether to enable TLS encryption automatically if a server supports it,
 * even if `$SMTPSecure` is not set to 'tls'.
 * Be aware that in PHP >= 5.6 this requires that the server's certificates are
valid.
*
* @var bool
*/
public $SMTPAutoTLS = true;

/** 
 * Whether to use SMTP authentication.
 * Uses the Username and Password properties.
*
* @see PHPMailer::$Username
* @see PHPMailer::$Password
*
* @var bool
*/
public $SMTPAuth = false;

/** 
 * Options array passed to stream_context_create when connecting via SMTP.
*
* @var array
*/
public $SMTPOptions = [];

/** 
 * SMTP username.
*
* @var string
*/
public $Username = "";

/** 
 * SMTP password.
*

```

```

* @var string
*/
public $Password = "";

/**
 * SMTP authentication type. Options are CRAM-MD5, LOGIN, PLAIN,
XOAUTH2.
 * If not specified, the first one from that list that the server supports will be
selected.
*
* @var string
*/
public $AuthType = "";

/**
 * SMTP SMTPXClient command attributes
*
* @var array
*/
protected $SMTPXClient = [];

/**
 * An implementation of the PHPMailer OAuthTokenProvider interface.
*
* @var OAuthTokenProvider
*/
protected $oauth;

/**
 * The SMTP server timeout in seconds.
* Default of 5 minutes (300sec) is from RFC2821 section 4.5.3.2.
*
* @var int
*/
public $Timeout = 300;

/**
 * Comma separated list of DSN notifications
* 'NEVER' under no circumstances a DSN must be returned to the sender.
*      If you use NEVER all other notifications will be ignored.
* 'SUCCESS' will notify you when your mail has arrived at its destination.
* 'FAILURE' will arrive if an error occurred during delivery.

```

```

* 'DELAY' will notify you if there is an unusual delay in delivery, but the actual
*       delivery's outcome (success or failure) is not yet decided.
*
*   * @see https://www.rfc-editor.org/rfc/rfc3461.html#section-4.1 for more
information about NOTIFY
*/
public $dsn = "";

/**
 * SMTP class debug output mode.
* Debug output level.
* Options:
* @see SMTP::DEBUG_OFF: No output
* @see SMTP::DEBUG_CLIENT: Client messages
* @see SMTP::DEBUG_SERVER: Client and server messages
* @see SMTP::DEBUG_CONNECTION: As SERVER plus connection status
* @see SMTP::DEBUG_LOWLEVEL: Noisy, low-level data output, rarely
needed
*
* @see SMTP::$do_debug
*
* @var int
*/
public $SMTPDebug = 0;

/**
* How to handle debug output.
* Options:
* * `echo` Output plain-text as-is, appropriate for CLI
* * `html` Output escaped, line breaks converted to `<br>`, appropriate for
browser output
* * `error_log` Output to error log as configured in php.ini
* By default PHPMailer will use `echo` if run from a `cli` or `cli-server` SAPI,
`html` otherwise.
* Alternatively, you can provide a callable expecting two params: a message
string and the debug level:
*
* ``php
* $mail->Debugoutput = function($str, $level) {echo "debug level $level;
message: $str";};
*
*

```

```

    * Alternatively, you can pass in an instance of a PSR-3 compatible logger,
    though only `debug`
    * level output is used:
    *
    * ``php
    * $mail->Debugoutput = new myPsr3Logger;
    * ``
    *
    * @see SMTP::$Debugoutput
    *
    * @var string|callable|\Psr\Log\LoggerInterface
    */
public $Debugoutput = 'echo';

/**
 * Whether to keep the SMTP connection open after each message.
 * If this is set to true then the connection will remain open after a send,
 * and closing the connection will require an explicit call to smtpClose().
 * It's a good idea to use this if you are sending multiple messages as it reduces
overhead.
 * See the mailing list example for how to use it.
 *
 * @var bool
*/
public $SMTPKeepAlive = false;

/**
 * Whether to split multiple to addresses into multiple messages
 * or send them all in one message.
 * Only supported in `mail` and `sendmail` transports, not in SMTP.
 *
 * @var bool
*
 * @deprecated 6.0.0 PHPMailer isn't a mailing list manager!
*/
public $SingleTo = false;

/**
 * Storage for addresses when SingleTo is enabled.
 *
 * @var array
*/

```

```

protected $SingleToArray = [];

/**
 * Whether to generate VERP addresses on send.
 * Only applicable when sending via SMTP.
 *
 * @see https://en.wikipedia.org/wiki/Variable_envelope_return_path
 * @see https://www.postfix.org/VERP_README.html Postfix VERP info
 *
 * @var bool
 */
public $do_verp = false;

/**
 * Whether to allow sending messages with an empty body.
 *
 * @var bool
 */
public $AllowEmpty = false;

/**
 * DKIM selector.
 *
 * @var string
 */
public $DKIM_selector = "";

/**
 * DKIM Identity.
 * Usually the email address used as the source of the email.
 *
 * @var string
 */
public $DKIM_identity = "";

/**
 * DKIM passphrase.
 * Used if your key is encrypted.
 *
 * @var string
 */
public $DKIM_passphrase = "";

```

```

/***
 * DKIM signing domain name.
 *
 * @example 'example.com'
 *
 * @var string
 */
public $DKIM_domain = "";

/***
 * DKIM Copy header field values for diagnostic use.
 *
 * @var bool
 */
public $DKIM_copyHeaderFields = true;

/***
 * DKIM Extra signing headers.
 *
 * @example ['List-Unsubscribe', 'List-Help']
 *
 * @var array
 */
public $DKIM_extraHeaders = [];

/***
 * DKIM private key file path.
 *
 * @var string
 */
public $DKIM_private = "";

/***
 * DKIM private key string.
 *
 * If set, takes precedence over `'$DKIM_private`'.
 *
 * @var string
 */
public $DKIM_private_string = "";

```

```

/**
 * Callback Action function name.
 *
 * The function that handles the result of the send email action.
 * It is called out by send() for each email sent.
 *
 * Value can be any php callable: https://www.php.net/is_callable
 *
 * Parameters:
 *   bool $result      result of the send action
 *   array $to         email addresses of the recipients
 *   array $cc          cc email addresses
 *   array $bcc         bcc email addresses
 *   string $subject    the subject
 *   string $body        the email body
 *   string $from        email address of sender
 *   string $extra       extra information of possible use
 *                      "smtp_transaction_id" => last smtp transaction id
 *
 * @var string
 */
public $action_function = '';

/**
 * What to put in the X-Mailer header.
 * Options: An empty string for PHPMailer default, whitespace/null for none, or
 * a string to use.
 *
 * @var string|null
 */
public $XMailer = '';

/**
 * Which validator to use by default when validating email addresses.
 * May be a callable to inject your own validator, but there are several built-in
 * validators.
 * The default validator uses PHP's FILTER_VALIDATE_EMAIL filter_var
 * option.
 *
 * If CharSet is UTF8, the validator is left at the default value,
 * and you send to addresses that use non-ASCII local parts, then
 * PHPMailer automatically changes to the 'eai' validator.

```

```

*
 * @see PHPMailer::validateAddress()
 *
 * @var string|callable
 */
public static $validator = 'php';

/**
 * An instance of the SMTP sender class.
 *
 * @var SMTP
 */
protected $smtp;

/**
 * The array of 'to' names and addresses.
 *
 * @var array
 */
protected $to = [];

/**
 * The array of 'cc' names and addresses.
 *
 * @var array
 */
protected $cc = [];

/**
 * The array of 'bcc' names and addresses.
 *
 * @var array
 */
protected $bcc = [];

/**
 * The array of reply-to names and addresses.
 *
 * @var array
 */
protected $ReplyTo = [];

```

```

/**
 * An array of all kinds of addresses.
 * Includes all of $to, $cc, $bcc.
 *
 * @see PHPMailer::$to
 * @see PHPMailer::$cc
 * @see PHPMailer::$bcc
 *
 * @var array
 */
protected $all_recipients = [];

/**
 * An array of names and addresses queued for validation.
 * In send(), valid and non duplicate entries are moved to $all_recipients
 * and one of $to, $cc, or $bcc.
 * This array is used only for addresses with IDN.
 *
 * @see PHPMailer::$to
 * @see PHPMailer::$cc
 * @see PHPMailer::$bcc
 * @see PHPMailer::$all_recipients
 *
 * @var array
*/
protected $RecipientsQueue = [];

/**
 * An array of reply-to names and addresses queued for validation.
 * In send(), valid and non duplicate entries are moved to $ReplyTo.
 * This array is used only for addresses with IDN.
 *
 * @see PHPMailer::$ReplyTo
 *
 * @var array
*/
protected $ReplyToQueue = [];

/**
 * Whether the need for SMTPUTF8 has been detected. Set by
 * preSend() if necessary.
 *

```

```

* @var bool
*/
public $UseSMTPUTF8 = false;

/** 
 * The array of attachments.
 *
 * @var array
 */
protected $attachment = [];

/** 
 * The array of custom headers.
 *
 * @var array
 */
protected $CustomHeader = [];

/** 
 * The most recent Message-ID (including angular brackets).
 *
 * @var string
 */
protected $lastMessageID = "";

/** 
 * The message's MIME type.
 *
 * @var string
 */
protected $message_type = "";

/** 
 * The array of MIME boundary strings.
 *
 * @var array
 */
protected $boundary = [];

/** 
 * The array of available text strings for the current language.
 *

```

```

* @var array
*/
protected $language = [];

/** 
 * The number of errors encountered.
 *
 * @var int
 */
protected $error_count = 0;

/** 
 * The S/MIME certificate file path.
 *
 * @var string
 */
protected $sign_cert_file = "";

/** 
 * The S/MIME key file path.
 *
 * @var string
 */
protected $sign_key_file = "";

/** 
 * The optional S/MIME extra certificates ("CA Chain") file path.
 *
 * @var string
 */
protected $sign_extracerts_file = "";

/** 
 * The S/MIME password for the key.
 * Used only if the key is encrypted.
 *
 * @var string
 */
protected $sign_key_pass = "";

/** 
 * Whether to throw exceptions for errors.

```

```

/*
 * @var bool
 */
protected $exceptions = false;

/**
 * Unique ID used for message ID and boundaries.
 *
 * @var string
 */
protected $uniqueid = "";

/**
 * The PHPMailer Version number.
 *
 * @var string
 */
const VERSION = '6.10.0';

/**
 * Error severity: message only, continue processing.
 *
 * @var int
 */
const STOP_MESSAGE = 0;

/**
 * Error severity: message, likely ok to continue processing.
 *
 * @var int
 */
const STOP_CONTINUE = 1;

/**
 * Error severity: message, plus full stop, critical error reached.
 *
 * @var int
 */
const STOP_CRITICAL = 2;

/**
 * The SMTP standard CRLF line break.

```

```

* If you want to change line break format, change static::$LE, not this.
*/
const CRLF = "\r\n";

/**
 * "Folding White Space" a white space string used for line folding.
 */
const FWS = ' ';

/**
 * SMTP RFC standard line ending; Carriage Return, Line Feed.
 *
 * @var string
 */
protected static $LE = self::CRLF;

/**
 * The maximum line length supported by mail().
 *
 * Background: mail() will sometimes corrupt messages
 * with headers longer than 65 chars, see #818.
 *
 * @var int
 */
const MAIL_MAX_LINE_LENGTH = 63;

/**
 * The maximum line length allowed by RFC 2822 section 2.1.1.
 *
 * @var int
 */
const MAX_LINE_LENGTH = 998;

/**
 * The lower maximum line length allowed by RFC 2822 section 2.1.1.
 * This length does NOT include the line break
 * 76 means that lines will be 77 or 78 chars depending on whether
 * the line break format is LF or CRLF; both are valid.
 *
 * @var int
 */
const STD_LINE_LENGTH = 76;

```

```

/**
 * Constructor.
 *
 * @param bool $exceptions Should we throw external exceptions?
 */
public function __construct($exceptions = null)
{
    if (null !== $exceptions) {
        $this->exceptions = (bool) $exceptions;
    }
    //Pick an appropriate debug output format automatically
    $this->Debugoutput = (strpos(PHP_SAPI, 'cli') !== false ? 'echo' : 'html');
}

/**
 * Destructor.
 */
public function __destruct()
{
    //Close any open SMTP connection nicely
    $this->smtpClose();
}

/**
 * Call mail() in a safe_mode-aware fashion.
 * Also, unless sendmail_path points to sendmail (or something that
 * claims to be sendmail), don't pass params (not a perfect fix,
 * but it will do).
 *
 * @param string    $to      To
 * @param string    $subject Subject
 * @param string    $body    Message Body
 * @param string    $header  Additional Header(s)
 * @param string|null $params Params
 *
 * @return bool
 */
private function mailPassthru($to, $subject, $body, $header, $params)
{
    //Check overloading of mail function to avoid double-encoding
    if ((int)ini_get('mbstring.func_overload') & 1) {

```

```

        $subject = $this->secureHeader($subject);
    } else {
        $subject = $this->encodeHeader($this->secureHeader($subject));
    }
    //Calling mail() with null params breaks
    $this->edebug('Sending with mail()');
    $this->edebug('Sendmail path: ' . ini_get('sendmail_path'));
    $this->edebug("Envelope sender: {$this->Sender}");
    $this->edebug("To: {$to}");
    $this->edebug("Subject: {$subject}");
    $this->edebug("Headers: {$header}");
    if (!$this->UseSendmailOptions || null === $params) {
        $result = @mail($to, $subject, $body, $header);
    } else {
        $this->edebug("Additional params: {$params}");
        $result = @mail($to, $subject, $body, $header, $params);
    }
    $this->edebug('Result: ' . ($result ? 'true' : 'false'));
    return $result;
}

/**
 * Output debugging info via a user-defined method.
 * Only generates output if debug output is enabled.
 *
 * @see PHPMailer::$Debugoutput
 * @see PHPMailer::$SMTPDebug
 *
 * @param string $str
 */
protected function edebug($str)
{
    if ($this->SMTPDebug <= 0) {
        return;
    }
    //Is this a PSR-3 logger?
    if ($this->Debugoutput instanceof \Psr\Log\LoggerInterface) {
        $this->Debugoutput->debug(rtrim($str, "\r\n"));
    }
    return;
}
//Avoid clash with built-in function names

```

```

    if (is_callable($this->Debugoutput) && !in_array($this->Debugoutput,
['error_log', 'html', 'echo'])) {
        call_user_func($this->Debugoutput, $str, $this->SMTPDebug);

        return;
    }
    switch ($this->Debugoutput) {
        case 'error_log':
            //Don't output, just log
            /** @noinspection ForgottenDebugOutputInspection */
            error_log($str);
            break;
        case 'html':
            //Cleans up output a bit for a better looking, HTML-safe output
            echo htmlentities(
                preg_replace('/[\r\n]+/', " ", $str),
                ENT_QUOTES,
                'UTF-8'
            ), "<br>\n";
            break;
        case 'echo':
        default:
            //Normalize line breaks
            $str = preg_replace('/\r\n|\r/m', "\n", $str);
            echo gmdate('Y-m-d H:i:s'),
            "\t",
            //Trim trailing space
            trim(
                //Indent for readability, except for trailing break
                str_replace(
                    "\n",
                    "\n\t",
                    trim($str)
                )
            ),
            "\n";
    }
}

/**
 * Sets message type to HTML or plain.
 *

```

```

* @param bool $isHtml True for HTML mode
*/
public function isHTML($isHtml = true)
{
    if ($isHtml) {
        $this->ContentType = static::CONTENT_TYPE_TEXT_HTML;
    } else {
        $this->ContentType = static::CONTENT_TYPE_PLAINTEXT;
    }
}

/**
 * Send messages using SMTP.
 */
public function isSMTP()
{
    $this->Mailer = 'smtp';
}

/**
 * Send messages using PHP's mail() function.
 */
public function isMail()
{
    $this->Mailer = 'mail';
}

/**
 * Send messages using $Sendmail.
 */
public function isSendmail()
{
    $ini_sendmail_path = ini_get('sendmail_path');

    if (false === strpos($ini_sendmail_path, 'sendmail')) {
        $this->Sendmail = '/usr/sbin/sendmail';
    } else {
        $this->Sendmail = $ini_sendmail_path;
    }
    $this->Mailer = 'sendmail';
}

```

```

/**
 * Send messages using qmail.
 */
public function isQmail()
{
    $ini_sendmail_path = ini_get('sendmail_path');

    if (false === strpos($ini_sendmail_path, 'qmail')) {
        $this->Sendmail = '/var/qmail/bin/qmail-inject';
    } else {
        $this->Sendmail = $ini_sendmail_path;
    }
    $this->Mailer = 'qmail';
}

/**
 * Add a "To" address.
 *
 * @param string $address The email address to send to
 * @param string $name
 *
 * @throws Exception
 *
 * @return bool true on success, false if address already used or invalid in
 * some way
 */
public function addAddress($address, $name = "")
{
    return $this->addOrEnqueueAnAddress('to', $address, $name);
}

/**
 * Add a "CC" address.
 *
 * @param string $address The email address to send to
 * @param string $name
 *
 * @throws Exception
 *
 * @return bool true on success, false if address already used or invalid in
 * some way
 */

```

```

public function addCC($address, $name = "")
{
    return $this->addOrEnqueueAnAddress('cc', $address, $name);
}

/** 
 * Add a "BCC" address.
 *
 * @param string $address The email address to send to
 * @param string $name
 *
 * @throws Exception
 *
 * @return bool true on success, false if address already used or invalid in
some way
*/
public function addBCC($address, $name = "")
{
    return $this->addOrEnqueueAnAddress('bcc', $address, $name);
}

/** 
 * Add a "Reply-To" address.
 *
 * @param string $address The email address to reply to
 * @param string $name
 *
 * @throws Exception
 *
 * @return bool true on success, false if address already used or invalid in
some way
*/
public function addReplyTo($address, $name = "")
{
    return $this->addOrEnqueueAnAddress('Reply-To', $address, $name);
}

/** 
 * Add an address to one of the recipient arrays or to the ReplyTo array.
Because PHPMailer
    * can't validate addresses with an IDN without knowing the
PHPMailer::$CharSet (that can still

```

```

* be modified after calling this function), addition of such addresses is delayed
until send()).
* Addresses that have been added already return false, but do not throw
exceptions.
*
* @param string $kind One of 'to', 'cc', 'bcc', or 'Reply-To'
* @param string $address The email address
* @param string $name An optional username associated with the address
*
* @throws Exception
*
* @return bool true on success, false if address already used or invalid in
some way
*/
protected function addOrEnqueueAnAddress($kind, $address, $name)
{
    $pos = false;
    if ($address !== null) {
        $address = trim($address);
        $pos = strpos($address, '@');
    }
    if (false === $pos) {
        //At-sign is missing.
        $error_message = sprintf(
            '%s (%s): %s',
            $this->lang('invalid_address'),
            $kind,
            $address
        );
        $this->setError($error_message);
        $this->edebug($error_message);
        if ($this->exceptions) {
            throw new Exception($error_message);
        }
    }

    return false;
}
if ($name !== null && is_string($name)) {
    $name = trim(preg_replace('/[\r\n]+/', " ", $name)); //Strip breaks and trim
} else {
    $name = "";
}

```

```

$params = [$kind, $address, $name];
//Enqueue addresses with IDN until we know the PHPMailer::$CharSet.
//Domain is assumed to be whatever is after the last @ symbol in the
address
if ($this->has8bitChars(substr($address, ++$pos))) {
    if (static::idnSupported()) {
        if ('Reply-To' !== $kind) {
            if (!array_key_exists($address, $this->RecipientsQueue)) {
                $this->RecipientsQueue[$address] = $params;

                    return true;
            }
        } elseif (!array_key_exists($address, $this->ReplyToQueue)) {
            $this->ReplyToQueue[$address] = $params;

                return true;
        }
    }
}

//We have an 8-bit domain, but we are missing the necessary extensions
to support it
//Or we are already sending to this address
return false;
}

//Immediately add standard addresses without IDN.
return call_user_func_array([$this, 'addAnAddress'], $params);
}

/**
 * Set the boundaries to use for delimiting MIME parts.
 * If you override this, ensure you set all 3 boundaries to unique values.
 * The default boundaries include a "=_" sequence which cannot occur in
quoted-printable bodies,
 * as suggested by https://www.rfc-editor.org/rfc/rfc2045#section-6.7
 *
 * @return void
 */
public function setBoundaries()
{
    $this->uniqueid = $this->generateId();
    $this->boundary[1] = 'b1=_'. $this->uniqueid;
    $this->boundary[2] = 'b2=_'. $this->uniqueid;
}

```

```

    $this->boundary[3] = 'b3=_'. $this->uniqueid;
}

/** 
 * Add an address to one of the recipient arrays or to the ReplyTo array.
 * Addresses that have been added already return false, but do not throw
exceptions.
*
* @param string $kind One of 'to', 'cc', 'bcc', or 'ReplyTo'
* @param string $address The email address to send, resp. to reply to
* @param string $name
*
* @throws Exception
*
* @return bool true on success, false if address already used or invalid in
some way
*/
protected function addAnAddress($kind, $address, $name = "")
{
    if (
        self::$validator === 'php' &&
        ((bool) preg_match('/[\x80-\xFF]/', $address))
    ) {
        //The caller has not altered the validator and is sending to an address
        //with UTF-8, so assume that they want UTF-8 support instead of failing
        $this->CharSet = self::CHARSET_UTF8;
        self::$validator = 'eai';
    }
    if (!in_array($kind, ['to', 'cc', 'bcc', 'Reply-To'])) {
        $error_message = sprintf(
            '%s: %s',
            $this->lang('Invalid recipient kind'),
            $kind
        );
        $this->setError($error_message);
        $this->edebug($error_message);
        if ($this->exceptions) {
            throw new Exception($error_message);
        }
    }
    return false;
}

```

```

if (!static::validateAddress($address)) {
    $error_message = sprintf(
        '%s (%s): %s',
        $this->lang('invalid_address'),
        $kind,
        $address
    );
    $this->setError($error_message);
    $this->edebug($error_message);
    if ($this->exceptions) {
        throw new Exception($error_message);
    }

    return false;
}

if ('Reply-To' !== $kind) {
    if (!array_key_exists(strtolower($address), $this->all_recipients)) {
        $this->{$kind}[] = [$address, $name];
        $this->all_recipients[strtolower($address)] = true;

        return true;
    }
} elseif (!array_key_exists(strtolower($address), $this->ReplyTo)) {
    $this->ReplyTo[strtolower($address)] = [$address, $name];

    return true;
}

return false;
}

/**
 * Parse and validate a string containing one or more RFC822-style comma-separated email addresses
 * of the form "display name <address>" into an array of name/address pairs.
 * Uses the imap_rfc822_parse_adrlist function if the IMAP extension is available.
 * Note that quotes in the name part are removed.
 *
 * @see

```

<https://www.andrew.cmu.edu/user/agreen1/testing/mrbs/web/Mail/RFC822.php>

A more careful implementation

```

*
 * @param string $addrstr The address list string
 * @param bool   $useimap Whether to use the IMAP extension to parse the
list
 * @param string $charset The charset to use when decoding the address list
string.
*
* @return array
*/
public static function parseAddresses($addrstr, $useimap = true, $charset =
self::CHARSET_ISO88591)
{
    $addresses = [];
    if ($useimap && function_exists('imap_rfc822_parse_adrlist')) {
        //Use this built-in parser if it's available
        $list = imap_rfc822_parse_adrlist($addrstr, '');
        // Clear any potential IMAP errors to get rid of notices being thrown at end
of script.
        imap_errors();
        foreach ($list as $address) {
            if (
                '.SYNTAX-ERROR.' !== $address->host &&
                static::validateAddress($address->mailbox . '@' . $address->host)
            ) {
                //Decode the name part if it's present and encoded
                if (
                    property_exists($address, 'personal') &&
                    //Check for a Mbstring constant rather than using
extension_loaded, which is sometimes disabled
                    defined('MB_CASE_UPPER') &&
                    preg_match('/^=\\?.*\\?=$/s', $address->personal)
                ) {
                    $origCharset = mb_internal_encoding();
                    mb_internal_encoding($charset);
                    //Undo any RFC2047-encoded spaces-as-underscores
                    $address->personal = str_replace('_', '=20', $address-
>personal);
                    //Decode the name
                    $address->personal = mb_decode_mimeheader($address-
>personal);
                    mb_internal_encoding($origCharset);
                }
            }
        }
    }
}

```

```

$addresses[] = [
    'name' => (property_exists($address, 'personal') ? $address->personal : ''),
    'address' => $address->mailbox . '@' . $address->host,
];
}
}
} else {
//Use this simpler parser
$list = explode(',', $addrstr);
foreach ($list as $address) {
$address = trim($address);
//Is there a separate name part?
if (strpos($address, '<') === false) {
//No separate name, just use the whole thing
if (static::validateAddress($address)) {
$addresses[] = [
    'name' => '',
    'address' => $address,
];
}
} else {
list($name, $email) = explode('<', $address);
$email = trim(str_replace('>', '', $email));
$name = trim($name);
if (static::validateAddress($email)) {
//Check for a Mbstring constant rather than using
extension_loaded, which is sometimes disabled
//If this name is encoded, decode it
if (defined('MB_CASE_UPPER') && preg_match('/^=|=.*\?=$/s',
$name)) {
$origCharset = mb_internal_encoding();
mb_internal_encoding($charset);
//Undo any RFC2047-encoded spaces-as-underscores
$name = str_replace('_', '=20', $name);
//Decode the name
$name = mb_decode_mimeheader($name);
mb_internal_encoding($origCharset);
}
$addresses[] =
//Remove any surrounding quotes and spaces from the name

```

```

        'name' => trim($name, '" '),
        'address' => $email,
    ];
}
}
}
}

return $addresses;
}

/** 
 * Set the From and FromName properties.
 *
 * @param string $address
 * @param string $name
 * @param bool   $auto  Whether to also set the Sender address, defaults to
true
 *
 * @throws Exception
 *
 * @return bool
 */
public function setFrom($address, $name = "", $auto = true)
{
    $address = trim((string)$address);
    $name = trim(preg_replace('/[\r\n]+/', " ", $name)); //Strip breaks and trim
    //Don't validate now addresses with IDN. Will be done in send().
    $pos = strrpos($address, '@');
    if (
        (false === $pos)
        ||      (!(!$this->has8bitChars(substr($address,      ++$pos))      ||
!static::idnSupported())
        && !static::validateAddress($address))
    ) {
        $error_message = sprintf(
            '%s (From): %s',
            $this->lang('invalid_address'),
            $address
        );
        $this->setError($error_message);
        $this->edebug($error_message);
    }
}

```

```

if ($this->exceptions) {
    throw new Exception($error_message);
}

return false;
}

$this->From = $address;
$this->FromName = $name;
if ($auto && empty($this->Sender)) {
    $this->Sender = $address;
}

return true;
}

/** 
 * Return the Message-ID header of the last email.
 * Technically this is the value from the last time the headers were created,
 * but it's also the message ID of the last sent message except in
 * pathological cases.
 *
 * @return string
 */
public function getLastMessageID()
{
    return $this->lastMessageID;
}

/** 
 * Check that a string looks like an email address.
 * Validation patterns supported:
 * * `auto` Pick best pattern automatically;
 * * `pcre8` Use the squilople.com pattern, requires PCRE > 8.0;
 * * `pcre` Use old PCRE implementation;
 * * `php` Use PHP built-in FILTER_VALIDATE_EMAIL;
 * * `html5` Use the pattern given by the HTML5 spec for 'email' type form input
elements.
 * * `eai` Use a pattern similar to the HTML5 spec for 'email' and to firefox,
extended to support EAI (RFC6530).
 * * `noregex` Don't use a regex: super fast, really dumb.
 * Alternatively you may pass in a callable to inject your own validator, for
example:

```

```

*
* ``php
* PHPMailer::validateAddress('user@example.com', function($address) {
*     return (strpos($address, '@') !== false);
* });
* ``
*
* You can also set the PHPMailer::$validator static to a callable, allowing built-in methods to use your validator.
*
* @param string      $address      The email address to check
* @param string|callable $patternselect Which pattern to use
*
* @return bool
*/
public static function validateAddress($address, $patternselect = null)
{
    if (null === $patternselect) {
        $patternselect = static::$validator;
    }
    //Don't allow strings as callables, see SECURITY.md and CVE-2021-3603
    if (is_callable($patternselect) && !is_string($patternselect)) {
        return call_user_func($patternselect, $address);
    }
    //Reject line breaks in addresses; it's valid RFC5322, but not RFC5321
    if (strpos($address, "\n") !== false || strpos($address, "\r") !== false) {
        return false;
    }
    switch ($patternselect) {
        case 'pcre': //Kept for BC
        case 'pcre8':
            /*
             * A more complex and more permissive version of the RFC5322 regex
on which FILTER_VALIDATE_EMAIL
             * is based.
             * In addition to the addresses allowed by filter_var, also permits:
             * * dotless domains: `a@b`
             * * comments: `1234 @ local(blah) .machine .example`
             * * quoted elements: `""test blah"@example.org"`
             * * numeric TLDs: `a@b.123`
             * * unbracketed IPv4 literals: `a@192.168.0.1`
             * * IPv6 literals: 'first.last@[IPv6:a1::]'
```

```

* Not all of these will necessarily work for sending!
*
* @copyright 2009-2010 Michael Rushton
* Feel free to use and redistribute this code. But please keep this
copyright notice.

*/
return (bool) preg_match(
    '/^(?!(>(?1)"?(?>\\"[ -~]"|[^\"])"?(?1)){255,})(?!(>(?1)"?(?>\\"[ -~]"|[^\"])"?(?1)){65,}@').
        '((>(?>(?>((?>(?>\x0D\x0A)?[\t ])+|(?>[\t ]*\x0D\x0A)?[\t ]+)?)(\((?>(?2)' .
            '(?>[\x01-\x08\x0B\x0C\x0E-\'*-\[\]-\x7F]|\\"[\x00-\x7F]|(?3)))*(?2)\))+(?2))|(?2))?' .
            '[!#-\'*+\/-9=?^-\~]+|"(?>(?2)(?>[\x01-\x08\x0B\x0C\x0E-\#-\[\]-\x7F]|\\"[\x00-\x7F])|\\"[\x00-\x7F]))*' .
            '(?2")(?>(?1).(?1)(?4))*(?1)@(?!(?1)[a-z0-9-]{64,})(?1)(?>([a-z0-9](?>[a-z0-9-]*[a-z0-9])))' .
            '?(>(?1).(?!(?1)[a-z0-9-]{64,})(?1)(?5)){0,126}|[(?:(>IPv6:(?>([a-f0-9]{1,4})(?>:(?6)){7})' .
            '|(?!(?:*[a-f0-9][:\"]){8,})((?6)(?>:(?6)){0,6})?::(?7)))|(?>(?>IPv6:(?>(?6)(?>:(?6)){5}):' .
            '|(?!(?:*[a-f0-9]:){6,})(?8)?::(?>((?6)(?>:(?6)){0,4})):)?)(25[0-5]|2[0-4][0-9]|1[0-9]{2})' .
            '|[1-9]?[0-9])(?>\.(?9)){3})))])(?1)$/isD',
    $address
);
case 'html5':
/*
* This is the pattern used in the HTML5 spec for validation of 'email'
type form input elements.
*
* @see https://html.spec.whatwg.org/#e-mail-state-(type=email)
*/
return (bool) preg_match(
    '/^@[a-zA-Z0-9.!#$%&\'*+\=\?\^_\`{}]\~]+@[a-zA-Z0-9](?:[a-zA-Z0-9]{0,61}' .
        '[a-zA-Z0-9])?(?:\.[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?)?$/sD',
    $address
);
case 'eai':
/*

```

```

        * This is the pattern used in the HTML5 spec for validation of 'email'
type
        * form input elements (as above), modified to accept Unicode email
addresses.
        * This is also more lenient than Firefox' html5 spec, in order to make
the regex faster.
        * 'eai' is an acronym for Email Address Internationalization.
        * This validator is selected automatically if you attempt to use recipient
addresses
        * that contain Unicode characters in the local part.
        *
        * @see https://html.spec.whatwg.org/#e-mail-state-(type=email)
        * @see https://en.wikipedia.org/wiki/International_email
        */
return (bool) preg_match(
    '/^[-\p{L}\p{N}\p{M}!#$%&|^*+|=|^_`{|}{|~}+@[|\p{L}\p{N}\p{M}]({:[|\p{L}\p{N}\p{M}-]
{0,61}} .
    '[|\p{L}\p{N}\p{M}])?({:\.\[|\p{L}\p{N}\p{M}]} .
    '(?:[-|\p{L}\p{N}\p{M}]{0,61}[|\p{L}\p{N}\p{M}])?)*$$/usD',
    $address
);
case 'php':
default:
    return filter_var($address, FILTER_VALIDATE_EMAIL) !== false;
}
}

/**
 * Tells whether IDNs (Internationalized Domain Names) are supported or not.
This requires the
 * `intl` and `mbstring` PHP extensions.
 *
 * @return bool `true` if required functions for IDN support are present
 */
public static function idnSupported()
{
    return function_exists('idn_to_ascii') &&
function_exists('mb_convert_encoding');
}

/**

```

```

    * Converts IDN in given email address to its ASCII form, also known as
    punycode, if possible.
    * Important: Address must be passed in same encoding as currently set in
    PHPMailer:::$CharSet.
    * This function silently returns unmodified address if:
    * - No conversion is necessary (i.e. domain name is not an IDN, or is already
    in ASCII form)
    * - Conversion to punycode is impossible (e.g. required PHP functions are not
    available)
    * or fails for any reason (e.g. domain contains characters not allowed in an
    IDN).
    *
    * @see PHPMailer:::$CharSet
    *
    * @param string $address The email address to convert
    *
    * @return string The encoded address in ASCII form
    */
public function punyencodeAddress($address)
{
    //Verify we have required functions, CharSet, and at-sign.
    $pos = strrpos($address, '@');
    if (
        !empty($this->CharSet) &&
        false !== $pos &&
        static::idnSupported()
    ) {
        $domain = substr($address, ++$pos);
        //Verify CharSet string is a valid one, and domain properly encoded in
        this CharSet.
        if ($this->has8bitChars($domain) && @mb_check_encoding($domain,
        $this->CharSet)) {
            //Convert the domain from whatever charset it's in to UTF-8
            $domain = mb_convert_encoding($domain, self::CHARSET_UTF8,
            $this->CharSet);
            //Ignore IDE complaints about this line - method signature changed in
            PHP 5.4
            $errorcode = 0;
            if (defined('INTL_IDNA_VARIANT_UTS46')) {
                //Use the current punycode standard (appeared in PHP 7.2)
                $punycode = idn_to_ascii(
                    $domain,

```

```

        \|IDNA_DEFAULT      |      \|IDNA_USE_STD3_RULES      |
\|IDNA_CHECK_BIDI |                                |
                                \|IDNA_CHECK_CONTEXTJ      |
\|IDNA_NONTRANSITIONAL_TO_ASCII,
                                \|INTL_IDNA_VARIANT_UTS46
                                );
} elseif (defined('INTL_IDNA_VARIANT_2003')) {
    //Fall back to this old, deprecated/removed encoding
    $punycode      =      idn_to_ascii($domain,      $errorcode,
\|INTL_IDNA_VARIANT_2003);
} else {
    //Fall back to a default we don't know about
    $punycode = idn_to_ascii($domain, $errorcode);
}
if (false !== $punycode) {
    return substr($address, 0, $pos) . $punycode;
}
}

return $address;
}

/**
 * Create a message and send it.
 * Uses the sending method specified by $Mailer.
 *
 * @throws Exception
 *
 * @return bool false on error - See the ErrorInfo property for details of the
error
*/
public function send()
{
try {
if (!$this->preSend()) {
    return false;
}

return $this->postSend();
} catch (Exception $exc) {
    $this->mailHeader = ";

```

```

        $this->setError($exc->getMessage());
        if ($this->exceptions) {
            throw $exc;
        }

        return false;
    }
}

/***
 * Prepare a message for sending.
 *
 * @throws Exception
 *
 * @return bool
 */
public function preSend()
{
    if (
        'smtp' === $this->Mailer
        || ('mail' === $this->Mailer && (\PHP_VERSION_ID >= 80000 ||
stripos(PHP_OS, 'WIN') === 0))
    ) {
        //SMTP mandates RFC-compliant line endings
        //and it's also used with mail() on Windows
        static::setLE(self::CRLF);
    } else {
        //Maintain backward compatibility with legacy Linux command line
mailers
        static::setLE(PHP_EOL);
    }
    //Check for buggy PHP versions that add a header with an incorrect line
break
    if (
        'mail' === $this->Mailer
        && ((\PHP_VERSION_ID >= 70000 && \PHP_VERSION_ID < 70017)
        || (\PHP_VERSION_ID >= 70100 && \PHP_VERSION_ID < 70103))
        && ini_get('mail.add_x_header') === '1'
        && stripos(PHP_OS, 'WIN') === 0
    ) {
        trigger_error($this->lang('buggy_php'), E_USER_WARNING);
    }
}

```

```

try {
    $this->error_count = 0; //Reset errors
    $this->mailHeader = "";

    //The code below tries to support full use of Unicode,
    //while remaining compatible with legacy SMTP servers to
    //the greatest degree possible: If the message uses
    //Unicode in the local parts of any addresses, it is sent
    //using SMTPUTF8. If not, it is sent using
    //punycode-encoded domains and plain SMTP.
    if (
        static::CHARSET_UTF8 === strtolower($this->CharSet) &&
        ($this->anyAddressHasUnicodeLocalPart($this->RecipientsQueue) ||
         $this->anyAddressHasUnicodeLocalPart(array_keys($this-
>all_recipients)) ||
         $this->anyAddressHasUnicodeLocalPart($this->ReplyToQueue) ||
         $this->addressHasUnicodeLocalPart($this->From))
    ) {
        $this->UseSMTPUTF8 = true;
    }
    //Dequeue recipient and Reply-To addresses with IDN
    foreach (array_merge($this->RecipientsQueue, $this->ReplyToQueue)
as $params) {
        if (!$this->UseSMTPUTF8) {
            $params[1] = $this->punyencodeAddress($params[1]);
        }
        call_user_func_array([$this, 'addAnAddress'], $params);
    }
    if (count($this->to) + count($this->cc) + count($this->bcc) < 1) {
        throw new Exception($this->lang('provide_address'),
self::STOP_CRITICAL);
    }

    //Validate From, Sender, and ConfirmReadingTo addresses
    foreach(['From', 'Sender', 'ConfirmReadingTo'] as $address_kind) {
        if ($this->{$address_kind} === null) {
            $this->{$address_kind} = "";
            continue;
        }
        $this->{$address_kind} = trim($this->{$address_kind});
        if (empty($this->{$address_kind})) {

```

```

        continue;
    }
    $this->{$address_kind}      =      $this->punyencodeAddress($this-
>{$address_kind});
    if (!static::validateAddress($this->{$address_kind})) {
        $error_message = sprintf(
            '%s (%s): %s',
            $this->lang('invalid_address'),
            $address_kind,
            $this->{$address_kind}
        );
        $this->setError($error_message);
        $this->edebug($error_message);
        if ($this->exceptions) {
            throw new Exception($error_message);
        }
    }

    return false;
}
}

//Set whether the message is multipart/alternative
if ($this->alternativeExists()) {
    $this->ContentType = static::CONTENT_TYPE_MULTIPART_ALTERNATIVE;
}

$this->setMessageType();
//Refuse to send an empty message unless we are specifically allowing
it
if (!$this->AllowEmpty && empty($this->Body)) {
    throw new Exception($this->lang('empty_message'),
self::STOP_CRITICAL);
}

//Trim subject consistently
$this->Subject = trim($this->Subject);
//Create body before headers in case body makes changes to headers
(e.g. altering transfer encoding)
$this->MIMEHeader = "";
$this->MIMEBody = $this->createBody();
//createBody may have added some headers, so retain them

```

```

$tempheaders = $this->MIMEHeader;
$this->MIMEHeader = $this->createHeader();
$this->MIMEHeader .= $tempheaders;

//To capture the complete message when using mail(), create
//an extra header list which createHeader() doesn't fold in
if ('mail' === $this->Mailer) {
    if (count($this->to) > 0) {
        $this->mailHeader .= $this->addrAppend('To', $this->to);
    } else {
        $this->mailHeader .= $this->headerLine('To', 'undisclosed-
recipients:');
    }
    $this->mailHeader .= $this->headerLine(
        'Subject',
        $this->encodeHeader($this->secureHeader($this->Subject))
    );
}

//Sign with DKIM if enabled
if (
    !empty($this->DKIM_domain)
    && !empty($this->DKIM_selector)
    && (!empty($this->DKIM_private_string)
        || (!empty($this->DKIM_private)
            && static::isPermittedPath($this->DKIM_private)
            && file_exists($this->DKIM_private)
        )
    )
) {
    $header_dkim = $this->DKIM_Add(
        $this->MIMEHeader . $this->mailHeader,
        $this->encodeHeader($this->secureHeader($this->Subject)),
        $this->MIMEBody
    );
    $this->MIMEHeader = static::stripTrailingWSP($this->MIMEHeader) .
static::$LE .
        static::normalizeBreaks($header_dkim) . static::$LE;
}

return true;
} catch (Exception $exc) {

```

```

        $this->setError($exc->getMessage());
        if ($this->exceptions) {
            throw $exc;
        }

        return false;
    }
}

/***
 * Actually send a message via the selected mechanism.
 *
 * @throws Exception
 *
 * @return bool
 */
public function postSend()
{
    try {
        //Choose the mailer and send through it
        switch ($this->Mailer) {
            case 'sendmail':
            case 'qmail':
                return      $this->sendmailSend($this->MIMEHeader,      $this-
>MIMEBody);
            case 'smtp':
                return $this->smtpSend($this->MIMEHeader, $this->MIMEBody);
            case 'mail':
                return $this->mailSend($this->MIMEHeader, $this->MIMEBody);
            default:
                $sendMethod = $this->Mailer . 'Send';
                if (method_exists($this, $sendMethod)) {
                    return      $this->{$sendMethod}($this->MIMEHeader,      $this-
>MIMEBody);
                }

                return $this->mailSend($this->MIMEHeader, $this->MIMEBody);
        }
    } catch (Exception $exc) {
        $this->setError($exc->getMessage());
        $this->edebug($exc->getMessage());
    }
}

```

```

        if ($this->Mailer === 'smtp' && $this->SMTPKeepAlive == true && $this-
>smtp->connected()) {
            $this->smtp->reset();
        }
        if ($this->exceptions) {
            throw $exc;
        }
    }

    return false;
}

/**
 * Send mail using the $Sendmail program.
 *
 * @see PHPMailer::$Sendmail
 *
 * @param string $header The message headers
 * @param string $body The message body
 *
 * @throws Exception
 *
 * @return bool
 */
protected function sendmailSend($header, $body)
{
    if ($this->Mailer === 'qmail') {
        $this->edebug('Sending with qmail');
    } else {
        $this->edebug('Sending with sendmail');
    }
    $header = static::stripTrailingWSP($header) . static::$LE . static::$LE;
    //This sets the SMTP envelope sender which gets turned into a return-path
header by the receiver
    //A space after '-f' is optional, but there is a long history of its presence
    //causing problems, so we don't use one
    //Exim          docs: https://www.exim.org/exim-html-
current/doc/html/spec_html/ch-the_exim_command_line.html
    //Sendmail docs: https://www.sendmail.org/~ca/email/man/sendmail.html
    //Example problem: https://www.drupal.org/node/1057954

    //PHP 5.6 workaround
}

```

```

$sendmail_from_value = ini_get('sendmail_from');
if (empty($this->Sender) && !empty($sendmail_from_value)) {
    //PHP config has a sender address we can use
    $this->Sender = ini_get('sendmail_from');
}
//CVE-2016-10033, CVE-2016-10045: Don't pass -f if characters will be
escaped.
if (!empty($this->Sender) && static::validateAddress($this->Sender) &&
self::isShellSafe($this->Sender)) {
    if ($this->Mailer === 'qmail') {
        $sendmailFmt = '%s -f%s';
    } else {
        $sendmailFmt = '%s -oi -f%s -t';
    }
} else {
    //allow sendmail to choose a default envelope sender. It may
    //seem preferable to force it to use the From header as with
    //SMTP, but that introduces new problems (see
    //<https://github.com/PHPMailer/PHPMailer/issues/2298>), and
    //it has historically worked this way.
    $sendmailFmt = '%s -oi -t';
}

$sendmail = sprintf($sendmailFmt, escapeshellcmd($this->Sendmail),
$this->Sender);
$this->edebug('Sendmail path: ' . $this->Sendmail);
$this->edebug('Sendmail command: ' . $sendmail);
$this->edebug('Envelope sender: ' . $this->Sender);
$this->edebug("Headers: {$header}");

if ($this->SingleTo) {
    foreach ($this->SingleToArray as $toAddr) {
        $mail = @popen($sendmail, 'w');
        if (!$mail) {
            throw new Exception($this->lang('execute') . $this->Sendmail,
self::STOP_CRITICAL);
        }
        $this->edebug("To: {$toAddr}");
        fwrite($mail, 'To: ' . $toAddr . "\n");
        fwrite($mail, $header);
        fwrite($mail, $body);
        $result = pclose($mail);
    }
}

```

```

$addrinfo = static::parseAddresses($toAddr, true, $this->CharSet);
$this->doCallback(
    ($result === 0),
    [[&$addrinfo['address'], $addrinfo['name']]],
    $this->cc,
    $this->bcc,
    $this->Subject,
    $body,
    $this->From,
    []
);
$this->edebug("Result: " . ($result === 0 ? 'true' : 'false'));
if (0 !== $result) {
    throw new Exception($this->lang('execute') . $this->Sendmail,
self::STOP_CRITICAL);
}
}
} else {
    $mail = @popen($sendmail, 'w');
    if (!$mail) {
        throw new Exception($this->lang('execute') . $this->Sendmail,
self::STOP_CRITICAL);
    }
    fwrite($mail, $header);
    fwrite($mail, $body);
    $result = pclose($mail);
    $this->doCallback(
        ($result === 0),
        $this->to,
        $this->cc,
        $this->bcc,
        $this->Subject,
        $body,
        $this->From,
        []
    );
    $this->edebug("Result: " . ($result === 0 ? 'true' : 'false'));
    if (0 !== $result) {
        throw new Exception($this->lang('execute') . $this->Sendmail,
self::STOP_CRITICAL);
    }
}
}

```

```

        return true;
    }

    /**
     * Fix CVE-2016-10033 and CVE-2016-10045 by disallowing potentially
     * unsafe shell characters.
     *
     * Note that escapeshellarg and escapeshellcmd are inadequate for our
     * purposes, especially on Windows.
     *
     * @see https://github.com/PHPMailer/PHPMailer/issues/924 CVE-2016-
     * 10045 bug report
     *
     * @param string $string The string to be validated
     *
     * @return bool
     */
protected static function isShellSafe($string)
{
    //It's not possible to use shell commands safely (which includes the mail()
    //function) without escapeshellarg,
    //but some hosting providers disable it, creating a security problem that we
    //don't want to have to deal with,
    //so we don't.
    if (!function_exists('escapeshellarg') || !function_exists('escapeshellcmd')) {
        return false;
    }

    if (
        escapeshellcmd($string) !== $string
        || !in_array(escapeshellarg($string), ["$string", "\"$string\""])
    ) {
        return false;
    }

    $length = strlen($string);

    for ($i = 0; $i < $length; ++$i) {
        $c = $string[$i];

        //All other characters have a special meaning in at least one common
        //shell, including = and +.
    }
}

```

//Full stop(.) has a special meaning in cmd.exe, but its impact should be negligible here.

//Note that this does permit non-Latin alphanumeric characters based on the current locale.

```
if (!ctype_alnum($c) && strpos('@_-.', $c) === false) {
    return false;
}

return true;
}

/** 
 * Check whether a file path is of a permitted type.
 * Used to reject URLs and phar files from functions that access local file paths,
 * such as addAttachment.
 *
 * @param string $path A relative or absolute path to a file
 *
 * @return bool
 */
protected static function isPermittedPath($path)
{
    //Matches      scheme      definition      from      https://www.rfc-
editor.org/rfc/rfc3986#section-3.1
    return !preg_match('#^([a-z][a-z\d+.-]*://#i', $path);
}

/** 
 * Check whether a file path is safe, accessible, and readable.
 *
 * @param string $path A relative or absolute path to a file
 *
 * @return bool
 */
protected static function fileIsAccessible($path)
{
    if (!static::isPermittedPath($path)) {
        return false;
    }
    $readable = is_file($path);
```

```

//If not a UNC path (expected to start with \\), check read permission, see
#2069
    if (strpos($path, '\\\\') !== 0) {
        $readable = $readable && is_readable($path);
    }
    return $readable;
}

/**
 * Send mail using the PHP mail() function.
 *
 * @see https://www.php.net/manual/en/book.mail.php
 *
 * @param string $header The message headers
 * @param string $body The message body
 *
 * @throws Exception
 *
 * @return bool
 */
protected function mailSend($header, $body)
{
    $header = static::stripTrailingWSP($header) . static::$LE . static::$LE;

    $toArr = [];
    foreach ($this->to as $toaddr) {
        $toArr[] = $this->addrFormat($toaddr);
    }
    $to = trim(implode(', ', $toArr));

    //If there are no To-addresses (e.g. when sending only to BCC-addresses)
    //the following should be added to get a correct DKIM-signature.
    //Compare with $this->preSend()
    if ($to === "") {
        $to = 'undisclosed-recipients:';
    }

    $params = null;
    //This sets the SMTP envelope sender which gets turned into a return-path
    header by the receiver
    //A space after '-f' is optional, but there is a long history of its presence
    //causing problems, so we don't use one

```

```

//Exim          docs: https://www.exim.org/exim-html-
current/doc/html/spec_html/ch-the_exim_command_line.html
//Sendmail docs: https://www.sendmail.org/~ca/email/man/sendmail.html
//Example problem: https://www.drupal.org/node/1057954
//CVE-2016-10033, CVE-2016-10045: Don't pass -f if characters will be
escaped.

//PHP 5.6 workaround
$sendmail_from_value = ini_get('sendmail_from');
if (empty($this->Sender) && !empty($sendmail_from_value)) {
    //PHP config has a sender address we can use
    $this->Sender = ini_get('sendmail_from');
}
if (!empty($this->Sender) && static::validateAddress($this->Sender)) {
    if (self::isShellSafe($this->Sender)) {
        $params = sprintf('-f%s', $this->Sender);
    }
    $old_from = ini_get('sendmail_from');
    ini_set('sendmail_from', $this->Sender);
}
$result = false;
if ($this->SingleTo && count($toArr) > 1) {
    foreach ($toArr as $toAddr) {
        $result = $this->mailPassthru($toAddr, $this->Subject, $body,
$header, $params);
        $addrinfo = static::parseAddresses($toAddr, true, $this->CharSet);
        $this->doCallback(
            $result,
            [[[$addrinfo['address'], $addrinfo['name']]],
            $this->cc,
            $this->bcc,
            $this->Subject,
            $body,
            $this->From,
            []
        );
    }
} else {
    $result = $this->mailPassthru($to, $this->Subject, $body, $header,
$params);
    $this->doCallback($result, $this->to, $this->cc, $this->bcc, $this-
>Subject, $body, $this->From, []);
}

```

```

        }

        if (isset($old_from)) {
            ini_set('sendmail_from', $old_from);
        }

        if (!$result) {
            throw new Exception($this->lang('instantiate'), self::STOP_CRITICAL);
        }

        return true;
    }

    /**
     * Get an instance to use for SMTP operations.
     * Override this function to load your own SMTP implementation,
     * or set one with setSMTPInstance.
     *
     * @return SMTP
     */
    public function getSMTPInstance()
    {
        if (!is_object($this->smtp)) {
            $this->smtp = new SMTP();
        }

        return $this->smtp;
    }

    /**
     * Provide an instance to use for SMTP operations.
     *
     * @return SMTP
     */
    public function setSMTPInstance(SMTP $smtp)
    {
        $this->smtp = $smtp;

        return $this->smtp;
    }

    /**
     * Provide SMTP XCLIENT attributes
     *

```

```

* @param string $name Attribute name
* @param ?string $value Attribute value
*
* @return bool
*/
public function setSMTPXclientAttribute($name, $value)
{
    if (!in_array($name, SMTP::$xclient_allowed_attributes)) {
        return false;
    }
    if (isset($this->SMTPXClient[$name]) && $value === null) {
        unset($this->SMTPXClient[$name]);
    } elseif ($value !== null) {
        $this->SMTPXClient[$name] = $value;
    }

    return true;
}

/**
* Get SMTP XCLIENT attributes
*
* @return array
*/
public function getSMTPXclientAttributes()
{
    return $this->SMTPXClient;
}

/**
* Send mail via SMTP.
* Returns false if there is a bad MAIL FROM, RCPT, or DATA input.
*
* @see PHPMailer::setSMTPInstance() to use a different class.
*
* @uses \PHPMailer\PHPMailer\SMTP
*
* @param string $header The message headers
* @param string $body The message body
*
* @throws Exception
*

```

```

* @return bool
*/
protected function smtpSend($header, $body)
{
    $header = static::stripTrailingWSP($header) . static::$LE . static::$LE;
    $bad_rcpt = [];
    if (!$this->smtpConnect($this->SMTPOptions)) {
        throw new Exception($this->lang('smtp_connect_failed'),
self::STOP_CRITICAL);
    }
    //If we have recipient addresses that need Unicode support,
    //but the server doesn't support it, stop here
    if ($this->UseSMTPUTF8 && !$this->smtp->getServerExt('SMTPUTF8')) {
        throw new Exception($this->lang('no_smtputf8'),
self::STOP_CRITICAL);
    }
    //Sender already validated in preSend()
    if (" === $this->Sender) {
        $smtp_from = $this->From;
    } else {
        $smtp_from = $this->Sender;
    }
    if (count($this->SMTPXClient)) {
        $this->smtp->xclient($this->SMTPXClient);
    }
    if (!$this->smtp->mail($smtp_from)) {
        $this->setError($this->lang('from_failed') . $smtp_from . ' : ' . implode(',',
$this->smtp->getError()));
        throw new Exception($this->ErrorInfo, self::STOP_CRITICAL);
    }

    $callbacks = [];
    //Attempt to send to all recipients
    foreach ([[$this->to, $this->cc, $this->bcc] as $togroup) {
        foreach ($togroup as $to) {
            if (!$this->smtp->recipient($to[0], $this->dsn)) {
                $error = $this->smtp->getError();
                $bad_rcpt[] = ['to' => $to[0], 'error' => $error['detail']];
                $isSent = false;
            } else {
                $isSent = true;
            }
        }
    }
}

```

```

        $callbacks[] = ['issent' => $isSent, 'to' => $to[0], 'name' => $to[1]];
    }
}

//Only send the DATA command if we have viable recipients
if ((count($this->all_recipients) > count($bad_rcpt)) && !$this->smtp->data($header . $body)) {
    throw new Exception($this->lang('data_not_accepted'),
self::STOP_CRITICAL);
}

$smtp_transaction_id = $this->smtp->getLastTransactionID();

if ($this->SMTPKeepAlive) {
    $this->smtp->reset();
} else {
    $this->smtp->quit();
    $this->smtp->close();
}

foreach ($callbacks as $cb) {
    $this->doCallback(
        $cb['issent'],
        [[[$cb['to'], $cb['name']]做人],
        [],
        [],
        $this->Subject,
        $body,
        $this->From,
        ['smtp_transaction_id' => $smtp_transaction_id]
    );
}

//Create error message for any bad addresses
if (count($bad_rcpt) > 0) {
    $errstr = "";
    foreach ($bad_rcpt as $bad) {
        $errstr .= $bad['to'] . ':' . $bad['error'];
    }
    throw new Exception($this->lang('recipients_failed') . $errstr,
self::STOP_CONTINUE);
}

```

```

        }

        return true;
    }

    /**
     * Initiate a connection to an SMTP server.
     * Returns false if the operation failed.
     *
     * @param array $options An array of options compatible with
     stream_context_create()
     *
     * @throws Exception
     *
     * @uses \PHPMailer\PHPMailer\SMTP
     *
     * @return bool
     */
    public function smtpConnect($options = null)
    {
        if (null === $this->smtp) {
            $this->smtp = $this->getSMTPInstance();
        }

        //If no options are provided, use whatever is set in the instance
        if (null === $options) {
            $options = $this->SMTPOptions;
        }

        //Already connected?
        if ($this->smtp->connected()) {
            return true;
        }

        $this->smtp->setTimeout($this->Timeout);
        $this->smtp->setDebugLevel($this->SMTPDebug);
        $this->smtp->setDebugOutput($this->Debugoutput);
        $this->smtp->setVerp($this->do_verp);
        $this->smtp->setSMTPUTF8($this->UseSMTPUTF8);
        if ($this->Host === null) {
            $this->Host = 'localhost';
        }
    }
}

```

```

$hosts = explode(';', $this->Host);
$lastexception = null;

foreach ($hosts as $hostentry) {
    $hostinfo = [];
    if (
        !preg_match(
            '/^(:?(ssl|tls):\/\/)?(.+?)(?:(\d+))?\$/',
            trim($hostentry),
            $hostinfo
        )
    ) {
        $this->edebug($this->lang('invalid_hostentry') . ' ' . trim($hostentry));
        //Not a valid host entry
        continue;
    }
    // $hostinfo[1]: optional ssl or tls prefix
    // $hostinfo[2]: the hostname
    // $hostinfo[3]: optional port number
    // The host string prefix can temporarily override the current setting for
    SMTPSecure
    // If it's not specified, the default value is used

    // Check the host name is a valid name or IP address before trying to use
    it
    if (!static::isValidHost($hostinfo[2])) {
        $this->edebug($this->lang('invalid_host') . ' ' . $hostinfo[2]);
        continue;
    }
    $prefix = '';
    $secure = $this->SMTPSecure;
    $tls = (static::ENCRYPTION_STARTTLS === $this->SMTPSecure);
    if ('ssl' === $hostinfo[1] || (" " === $hostinfo[1] &&
static::ENCRYPTION_SMTPS === $this->SMTPSecure)) {
        $prefix = 'ssl://';
        $tls = false; // Can't have SSL and TLS at the same time
        $secure = static::ENCRYPTION_SMTPS;
    } elseif ('tls' === $hostinfo[1]) {
        $tls = true;
        // TLS doesn't use a prefix
        $secure = static::ENCRYPTION_STARTTLS;
    }
}

```

```

//Do we need the OpenSSL extension?
$sslext = defined('OPENSSL_ALGO_SHA256');
if      (static::ENCRYPTION_STARTTLS     ===      $secure      ||
static::ENCRYPTION_SMTPS === $secure) {
    //Check for an OpenSSL constant rather than using extension_loaded,
which is sometimes disabled
    if (!$sslext) {
        throw new Exception($this->lang('extension_missing') . 'openssl',
self::STOP_CRITICAL);
    }
}
$host = $hostinfo[2];
$port = $this->Port;
if (
    array_key_exists(3, $hostinfo) &&
    is_numeric($hostinfo[3]) &&
    $hostinfo[3] > 0 &&
    $hostinfo[3] < 65536
) {
    $port = (int) $hostinfo[3];
}
if  ($this->smtp->connect($prefix . $host, $port, $this->Timeout,
$options)) {
    try {
        if ($this->Hello) {
            $hello = $this->Hello;
        } else {
            $hello = $this->serverHostname();
        }
        $this->smtp->hello($hello);
        //Automatically enable TLS encryption if:
        /* it's not disabled
        /* we are not connecting to localhost
        /* we have openssl extension
        /* we are not already using SSL
        /* the server offers STARTTLS
        if (
            $this->SMTPAutoTLS &&
            $this->Host !== 'localhost' &&
            $sslext &&
            $secure !== 'ssl' &&
            $this->smtp->getServerExt('STARTTLS')

```

```

    ) {
        $tls = true;
    }
    if ($tls) {
        if (!$this->smtp->startTLS()) {
            $message = $this->getSmtpErrorMessage('connect_host');
            throw new Exception($message);
        }
        //We must resend EHLO after TLS negotiation
        $this->smtp->hello($hello);
    }
    if (
        $this->SMTPAuth && !$this->smtp->authenticate(
            $this->Username,
            $this->Password,
            $this->AuthType,
            $this->oauth
        )
    ) {
        throw new Exception($this->lang('authenticate'));
    }

    return true;
} catch (Exception $exc) {
    $lastexception = $exc;
    $this->edebug($exc->getMessage());
    //We must have connected, but then failed TLS or Auth, so close
connection nicely
    $this->smtp->quit();
}
}
}

//If we get here, all connection attempts have failed, so close connection
hard
$this->smtp->close();
//As we've caught all exceptions, just report whatever the last one was
if ($this->exceptions && null !== $lastexception) {
    throw $lastexception;
}
if ($this->exceptions) {
    // no exception was thrown, likely $this->smtp->connect() failed
    $message = $this->getSmtpErrorMessage('connect_host');
}

```

```

        throw new Exception($message);
    }

    return false;
}

/** 
 * Close the active SMTP session if one exists.
 */
public function smtpClose()
{
    if ((null !== $this->smtp) && $this->smtp->connected()) {
        $this->smtp->quit();
        $this->smtp->close();
    }
}

/** 
 * Set the language for error messages.
 * The default language is English.
 *
 * @param string $langcode ISO 639-1 2-character language code (e.g.
French is "fr")
 *          Optionally, the language code can be enhanced with a 4-
character
 *          script annotation and/or a 2-character country annotation.
 * @param string $lang_path Path to the language file directory, with trailing
separator (slash)
 *          Do not set this from user input!
 *
 * @return bool Returns true if the requested language was loaded, false
otherwise.
*/
public function setLanguage($langcode = 'en', $lang_path = "")
{
    //Backwards compatibility for renamed language codes
    $renamed_langcodes = [
        'br' => 'pt_br',
        'cz' => 'cs',
        'dk' => 'da',
        'no' => 'nb',
        'se' => 'sv',

```

```

'rs' => 'sr',
'tg' => 'tl',
'am' => 'hy',
];

if (array_key_exists($langcode, $renamed_langcodes)) {
    $langcode = $renamed_langcodes[$langcode];
}

//Define full set of translatable strings in English
$PHPMAILER_LANG = [
    'authenticate' => 'SMTP Error: Could not authenticate.',
    'buggy_php' => 'Your version of PHP is affected by a bug that may result
in corrupted messages.' .
        ' To fix it, switch to sending using SMTP, disable the
mail.add_x_header option in'.
        ' your php.ini, switch to MacOS or Linux, or upgrade your PHP to
version 7.0.17+ or 7.1.3+'.,
    'connect_host' => 'SMTP Error: Could not connect to SMTP host.',
    'data_not_accepted' => 'SMTP Error: data not accepted.',
    'empty_message' => 'Message body empty',
    'encoding' => 'Unknown encoding: ',
    'execute' => 'Could not execute: ',
    'extension_missing' => 'Extension missing: ',
    'file_access' => 'Could not access file: ',
    'file_open' => 'File Error: Could not open file: ',
    'from_failed' => 'The following From address failed: ',
    'instantiate' => 'Could not instantiate mail function.',
    'invalid_address' => 'Invalid address: ',
    'invalid_header' => 'Invalid header name or value',
    'invalid_hostentry' => 'Invalid hostentry: ',
    'invalid_host' => 'Invalid host: ',
    'mailer_not_supported' => 'mailer is not supported.',
    'provide_address' => 'You must provide at least one recipient email
address.',

    'recipients_failed' => 'SMTP Error: The following recipients failed: ',
    'signing' => 'Signing Error: ',
    'smtp_code' => 'SMTP code: ',
    'smtp_code_ex' => 'Additional SMTP info: ',
    'smtp_connect_failed' => 'SMTP connect() failed.',
    'smtp_detail' => 'Detail: ',
    'smtp_error' => 'SMTP server error: ',

```

```

'variable_set' => 'Cannot set or reset variable: ',
'no_smtputf8' => 'Server does not support SMTPUTF8 needed to send
to Unicode addresses',
];

if (empty($lang_path)) {
    //Calculate an absolute path so it can work if CWD is not here
    $lang_path = dirname(__DIR__) . DIRECTORY_SEPARATOR .
    'language' . DIRECTORY_SEPARATOR;
}

//Validate $langcode
$foundlang = true;
$langcode = strtolower($langcode);
if (
    !preg_match('/^(?P<lang>[a-z]{2})(?P<script>_[a-
z]{4})?(?P<country>_[a-z]{2})?$/i', $langcode, $matches)
    && $langcode !== 'en'
) {
    $foundlang = false;
    $langcode = 'en';
}

//There is no English translation file
if ('en' !== $langcode) {
    $langcodes = [];
    if (!empty($matches['script']) && !empty($matches['country'])) {
        $langcodes[] = $matches['lang'] . $matches['script'] .
        $matches['country'];
    }
    if (!empty($matches['country'])) {
        $langcodes[] = $matches['lang'] . $matches['country'];
    }
    if (!empty($matches['script'])) {
        $langcodes[] = $matches['lang'] . $matches['script'];
    }
    $langcodes[] = $matches['lang'];

    //Try and find a readable language file for the requested language.
    $foundFile = false;
    foreach ($langcodes as $code) {
        $lang_file = $lang_path . 'phpmailer.lang-' . $code . '.php';
        if (static::fileIsAccessible($lang_file)) {

```

```

        $foundFile = true;
        break;
    }
}

if ($foundFile === false) {
    $foundLang = false;
} else {
    $lines = file($lang_file);
    foreach ($lines as $line) {
        //Translation file lines look like this:
        //${PHPMAILER_LANG['authenticate']}      =      'SMTP-Fehler:
Authentifizierung fehlgeschlagen.';

        //These files are parsed as text and not PHP so as to avoid the
        //possibility of code injection
        //See https://blog.stevenlevithan.com/archives/match-quoted-string
        $matches = [];
        if (
            preg_match(
                '/^${PHPMAILER_LANG}\[\'([a-zA-Z]+)\']\s*=|s*(["'])(.+)*?\2;/',
                $line,
                $matches
            ) &&
            //Ignore unknown translation keys
            array_key_exists($matches[1], ${PHPMAILER_LANG})
        ) {
            //Overwrite language-specific strings so we'll never have missing
            //translation keys.
            ${PHPMAILER_LANG}[$matches[1]] = (string)$matches[3];
        }
    }
}
$this->language = ${PHPMAILER_LANG};

return $foundLang; //Returns false if language not found
}

/**
 * Get the array of strings for the current language.
 *
 * @return array
 */

```

```

*/
public function getTranslations()
{
    if (empty($this->language)) {
        $this->setLanguage(); // Set the default language.
    }

    return $this->language;
}

/**
 * Create recipient headers.
 *
 * @param string $type
 * @param array $addr An array of recipients,
 *                   where each recipient is a 2-element indexed array with element
 * 0 containing an address
 *                   and element 1 containing a name, like:
 *                   [['joe@example.com', 'Joe User'], ['zoe@example.com', 'Zoe
User']]
 *
 * @return string
 */
public function addrAppend($type, $addr)
{
    $addresses = [];
    foreach ($addr as $address) {
        $addresses[] = $this->addrFormat($address);
    }

    return $type . ':' . implode(', ', $addresses) . static::$LE;
}

/**
 * Format an address for use in a message header.
 *
 * @param array $addr A 2-element indexed array, element 0 containing an
address, element 1 containing a name like
 *                   ['joe@example.com', 'Joe User']
 *
 * @return string
 */

```

```

public function addrFormat($addr)
{
    if (!isset($addr[1]) || ($addr[1] === "")) { //No name provided
        return $this->secureHeader($addr[0]);
    }

    return $this->encodeHeader($this->secureHeader($addr[1]), 'phrase') .
        '<' . $this->secureHeader($addr[0]) . '>';
}

/**
 * Word-wrap message.
 * For use with mailers that do not automatically perform wrapping
 * and for quoted-printable encoded messages.
 * Original written by philippe.
 *
 * @param string $message The message to wrap
 * @param int    $length  The line length to wrap to
 * @param bool   $qp_mode Whether to run in Quoted-Printable mode
 *
 * @return string
 */
public function wrapText($message, $length, $qp_mode = false)
{
    if ($qp_mode) {
        $soft_break = sprintf(' =%s', static::$LE);
    } else {
        $soft_break = static::$LE;
    }
    //If utf-8 encoding is used, we will need to make sure we don't
    //split multibyte characters when we wrap
    $is_utf8 = static::CHARSET_UTF8 === strtolower($this->CharSet);
    $lelen = strlen(static::$LE);
    $crlflen = strlen(static::$LE);

    $message = static::normalizeBreaks($message);
    //Remove a trailing line break
    if (substr($message, -$lelen) === static::$LE) {
        $message = substr($message, 0, -$lelen);
    }

    //Split message into lines

```

```

$lines = explode(static::$LE, $message);
//Message will be rebuilt in here
$message = "";
foreach ($lines as $line) {
    $words = explode(' ', $line);
    $buf = "";
    $firstword = true;
    foreach ($words as $word) {
        if ($qp_mode && (strlen($word) > $length)) {
            $space_left = $length - strlen($buf) - $crlflen;
            if (!$firstword) {
                if ($space_left > 20) {
                    $len = $space_left;
                    if ($is_utf8) {
                        $len = $this->utf8CharBoundary($word, $len);
                    } elseif ('=' === substr($word, $len - 1, 1)) {
                        --$len;
                    } elseif ('=' === substr($word, $len - 2, 1)) {
                        $len -= 2;
                    }
                    $part = substr($word, 0, $len);
                    $word = substr($word, $len);
                    $buf .= ' ' . $part;
                    $message .= $buf . sprintf('=%"s', static::$LE);
                } else {
                    $message .= $buf . $soft_break;
                }
                $buf = "";
            }
        while ($word !== "") {
            if ($length <= 0) {
                break;
            }
            $len = $length;
            if ($is_utf8) {
                $len = $this->utf8CharBoundary($word, $len);
            } elseif ('=' === substr($word, $len - 1, 1)) {
                --$len;
            } elseif ('=' === substr($word, $len - 2, 1)) {
                $len -= 2;
            }
            $part = substr($word, 0, $len);

```

```

        $word = (string) substr($word, $len);

        if ($word !== "") {
            $message .= $part . sprintf('=%s', static::$LE);
        } else {
            $buf = $part;
        }
    }
} else {
    $buf_o = $buf;
    if (!$firstword) {
        $buf .= ' ';
    }
    $buf .= $word;

    if ("!==" $buf_o && strlen($buf) > $length) {
        $message .= $buf_o . $soft_break;
        $buf = $word;
    }
}
$firstword = false;
}
$message .= $buf . static::$LE;
}

return $message;
}

/**
 * Find the last character boundary prior to $maxLength in a utf-8
 * quoted-printable encoded string.
 * Original written by Colin Brown.
 *
 * @param string $encodedText utf-8 QP text
 * @param int    $maxLength   Find the last character boundary prior to this
length
 *
 * @return int
 */
public function utf8CharBoundary($encodedText, $maxLength)
{
    $foundSplitPos = false;

```

```

$lookBack = 3;
while (!$foundSplitPos) {
    $lastChunk = substr($encodedText, $maxLength - $lookBack,
$lookBack);
    $encodedCharPos = strpos($lastChunk, '=');
    if (false !== $encodedCharPos) {
        //Found start of encoded character byte within $lookBack block.
        //Check the encoded byte value (the 2 chars after the '=')
        $hex = substr($encodedText, $maxLength - $lookBack +
$encodedCharPos + 1, 2);
        $dec = hexdec($hex);
        if ($dec < 128) {
            //Single byte character.
            //If the encoded char was found at pos 0, it will fit
            //otherwise reduce maxLength to start of the encoded char
            if ($encodedCharPos > 0) {
                $maxLength -= $lookBack - $encodedCharPos;
            }
            $foundSplitPos = true;
        } elseif ($dec >= 192) {
            //First byte of a multi byte character
            //Reduce maxLength to split at start of character
            $maxLength -= $lookBack - $encodedCharPos;
            $foundSplitPos = true;
        } elseif ($dec < 192) {
            //Middle byte of a multi byte character, look further back
            $lookBack += 3;
        }
    } else {
        //No encoded character found
        $foundSplitPos = true;
    }
}

return $maxLength;
}

/**
 * Apply word wrapping to the message body.
 * Wraps the message body to the number of chars set in the WordWrap
property.

```

```

* You should only do this to plain-text bodies as wrapping HTML tags may
break them.
* This is called automatically by createBody(), so you don't need to call it
yourself.
*/
public function setWordWrap()
{
    if ($this->WordWrap < 1) {
        return;
    }

    switch ($this->message_type) {
        case 'alt':
        case 'alt_inline':
        case 'alt_attach':
        case 'alt_inline_attach':
            $this->AltBody = $this->wrapText($this->AltBody, $this->WordWrap);
            break;
        default:
            $this->Body = $this->wrapText($this->Body, $this->WordWrap);
            break;
    }
}

/**
 * Assemble message headers.
 *
 * @return string The assembled headers
 */
public function createHeader()
{
    $result = "";

    $result .= $this->headerLine('Date', " === $this->MessageDate ? "
self::rfcDate() : $this->MessageDate);

    //The To header is created automatically by mail(), so needs to be omitted
here
    if ('mail' !== $this->Mailer) {
        if ($this->SingleTo) {
            foreach ($this->to as $toaddr) {
                $this->SingleToArray[] = $this->addrFormat($toaddr);
            }
        }
    }
}

```

```

    }
} elseif (count($this->to) > 0) {
    $result .= $this->addrAppend('To', $this->to);
} elseif (count($this->cc) === 0) {
    $result .= $this->headerLine('To', 'undisclosed-recipients:');
}
}

$result .= $this->addrAppend('From', [[trim($this->From), $this-
>FromName]]);

//sendmail and mail() extract Cc from the header before sending
if (count($this->cc) > 0) {
    $result .= $this->addrAppend('Cc', $this->cc);
}

//sendmail and mail() extract Bcc from the header before sending
if (
(
    'sendmail' === $this->Mailer || 'qmail' === $this->Mailer || 'mail' ===
$this->Mailer
)
&& count($this->bcc) > 0
) {
    $result .= $this->addrAppend('Bcc', $this->bcc);
}

if (count($this->ReplyTo) > 0) {
    $result .= $this->addrAppend('Reply-To', $this->ReplyTo);
}

//mail() sets the subject itself
if ('mail' !== $this->Mailer) {
    $result .= $this->headerLine('Subject', $this->encodeHeader($this-
>secureHeader($this->Subject)));
}

//Only allow a custom message ID if it conforms to RFC 5322 section 3.6.4
//https://www.rfc-editor.org/rfc/rfc5322#section-3.6.4
if (
    " !== $this->MessageID &&
preg_match(
    '/^<(((a-z\d!#$%&|^+|=?)^_`{|}~-]+(\.[a-z\d!#$%&|^+|=?)^_`{|}~-]+)*' .

```

```

'|("(([\\x01-\\x08\\x0B\\x0C\\x0E-\\x1F\\x7F])|([\\x21\\x23-\\x5B\\x5D-\\x7E])' .
'|(\\\\([\\x01-\\x09\\x0B\\x0C\\x0E-\\x7F]))*)")@(([a-z\\d!#$%&\\'*+\\/=?^_`{|}~-]+'.
`(.\\[a-z\\d!#$%&\\'*+\\/=?^_`{|}~-]+)*))|(\\\(([\\x01-\\x08\\x0B\\x0C\\x0E-
\\x1F\\x7F]' .
`|[\\x21-\\x5A\\x5E-\\x7E])|(\\\([\\x01-\\x09\\x0B\\x0C\\x0E-\\x7F]))*\'])))))>$/Di',
$this->MessageID
)
{
    $this->lastMessageID = $this->MessageID;
} else {
    $this->lastMessageID = sprintf('<%s@%s>', $this->uniqueid, $this-
>serverHostname());
}
$result .= $this->headerLine('Message-ID', $this->lastMessageID);
if (null !== $this->Priority) {
    $result .= $this->headerLine('X-Priority', $this->Priority);
}
if (" === $this->XMailer) {
    //Empty string for default X-Mailer header
    $result .= $this->headerLine(
        'X-Mailer',
        'PHPMailer' . self::VERSION . '
(https://github.com/PHPMailer/PHPMailer)'
    );
} elseif (is_string($this->XMailer) && trim($this->XMailer) !== "") {
    //Some string
    $result .= $this->headerLine('X-Mailer', trim($this->XMailer));
} //Other values result in no X-Mailer header

if (" != $this->ConfirmReadingTo) {
    $result .= $this->headerLine('Disposition-Notification-To', '<' . $this-
>ConfirmReadingTo . '>');
}

//Add custom headers
foreach ($this->CustomHeader as $header) {
    $result .= $this->headerLine(
        trim($header[0]),
        $this->encodeHeader(trim($header[1]))
    );
}

```

```

if (!$this->sign_key_file) {
    $result .= $this->headerLine('MIME-Version', '1.0');
    $result .= $this->getMailMIME();
}

return $result;
}

/** 
 * Get the message MIME type headers.
 *
 * @return string
 */
public function getMailMIME()
{
    $result = "";
    $ismultipart = true;
    switch ($this->message_type) {
        case 'inline':
            $result .= $this->headerLine('Content-Type',
static::CONTENT_TYPE_MULTIPART_RELATED . ';');
            $result .= $this->textLine(' boundary="" . $this->boundary[1] . ""');
            break;
        case 'attach':
        case 'inline_attach':
        case 'alt_attach':
        case 'alt_inline_attach':
            $result .= $this->headerLine('Content-Type',
static::CONTENT_TYPE_MULTIPART_MIXED . ';');
            $result .= $this->textLine(' boundary="" . $this->boundary[1] . ""');
            break;
        case 'alt':
        case 'alt_inline':
            $result .= $this->headerLine('Content-Type',
static::CONTENT_TYPE_MULTIPART_ALTERNATIVE . ';');
            $result .= $this->textLine(' boundary="" . $this->boundary[1] . ""');
            break;
        default:
            //Catches case 'plain': and case "":
            $result .= $this->textLine('Content-Type: ' . $this->ContentType . ';
charset=' . $this->CharSet);
            $ismultipart = false;
    }
}

```

```

        break;
    }
    //RFC1341 part 5 says 7bit is assumed if not specified
    if (static::ENCODING_7BIT != $this->Encoding) {
        //RFC 2045 section 6.4 says multipart MIME parts may only use 7bit, 8bit
        or binary CTE
        if ($ismultipart) {
            if (static::ENCODING_8BIT === $this->Encoding) {
                $result .= $this->headerLine('Content-Transfer-Encoding',
static::ENCODING_8BIT);
            }
            //The only remaining alternatives are quoted-printable and base64,
            which are both 7bit compatible
        } else {
            $result .= $this->headerLine('Content-Transfer-Encoding', $this-
>Encoding);
        }
    }

    return $result;
}

/**
 * Returns the whole MIME message.
 * Includes complete headers and body.
 * Only valid post preSend().
 *
 * @see PHPMailer::preSend()
 *
 * @return string
 */
public function getSentMIMEMessage()
{
    return static::stripTrailingWSP($this->MIMEHeader . $this->mailHeader) .
        static::$LE . static::$LE . $this->MIMEBody;
}

/**
 * Create a unique ID to use for boundaries.
 *
 * @return string
 */

```

```

protected function generateId()
{
    $len = 32; //32 bytes = 256 bits
    $bytes = "";
    if (function_exists('random_bytes')) {
        try {
            $bytes = random_bytes($len);
        } catch (\Exception $e) {
            //Do nothing
        }
    } elseif (function_exists('openssl_random_pseudo_bytes')) {
        /** @noinspection CryptographicallySecureRandomnessInspection */
        $bytes = openssl_random_pseudo_bytes($len);
    }
    if ($bytes === "") {
        //We failed to produce a proper random string, so make do.
        //Use a hash to force the length to the same as the other methods
        $bytes = hash('sha256', uniqid((string) mt_rand(), true), true);
    }

    //We don't care about messing up base64 format here, just want a random
    string
    return str_replace(['=', '+', '/'], " ", base64_encode(hash('sha256', $bytes,
true)));
}

/**
 * Assemble the message body.
 * Returns an empty string on failure.
 *
 * @throws Exception
 *
 * @return string The assembled message body
 */
public function createBody()
{
    $body = "";
    //Create unique IDs and preset boundaries
    $this->setBoundaries();

    if ($this->sign_key_file) {
        $body .= $this->getMailMIME() . static::$LE;

```

```

}

$this->setWordWrap();

$bodyEncoding = $this->Encoding;
$bodyCharSet = $this->CharSet;
//Can we do a 7-bit downgrade?
if ($this->UseSMTPUTF8) {
    $bodyEncoding = static::ENCODING_8BIT;
} elseif (static::ENCODING_8BIT === $bodyEncoding && !$this->has8bitChars($this->Body)) {
    $bodyEncoding = static::ENCODING_7BIT;
    //All ISO 8859, Windows codepage and UTF-8 charsets are ascii compatible up to 7-bit
    $bodyCharSet = static::CHARSET_ASCII;
}
//If lines are too long, and we're not already using an encoding that will shorten them,
//change to quoted-printable transfer encoding for the body part only
if (static::ENCODING_BASE64 !== $this->Encoding && static::hasLineLongerThanMax($this->Body)) {
    $bodyEncoding = static::ENCODING_QUOTED_PRINTABLE;
}

$altBodyEncoding = $this->Encoding;
$altBodyCharSet = $this->CharSet;
//Can we do a 7-bit downgrade?
if (static::ENCODING_8BIT === $altBodyEncoding && !$this->has8bitChars($this->AltBody)) {
    $altBodyEncoding = static::ENCODING_7BIT;
    //All ISO 8859, Windows codepage and UTF-8 charsets are ascii compatible up to 7-bit
    $altBodyCharSet = static::CHARSET_ASCII;
}
//If lines are too long, and we're not already using an encoding that will shorten them,
//change to quoted-printable transfer encoding for the alt body part only
if (static::ENCODING_BASE64 !== $altBodyEncoding && static::hasLineLongerThanMax($this->AltBody)) {
    $altBodyEncoding = static::ENCODING_QUOTED_PRINTABLE;
}
//Use this as a preamble in all multipart message types

```

```

$mimepre = "";
switch ($this->message_type) {
    case 'inline':
        $body .= $mimepre;
        $body .= $this->getBoundary($this->boundary[1], $bodyCharSet, ", $bodyEncoding);
        $body .= $this->encodeString($this->Body, $bodyEncoding);
        $body .= static::$LE;
        $body .= $this->attachAll('inline', $this->boundary[1]);
        break;
    case 'attach':
        $body .= $mimepre;
        $body .= $this->getBoundary($this->boundary[1], $bodyCharSet, ", $bodyEncoding);
        $body .= $this->encodeString($this->Body, $bodyEncoding);
        $body .= static::$LE;
        $body .= $this->attachAll('attachment', $this->boundary[1]);
        break;
    case 'inline_attach':
        $body .= $mimepre;
        $body .= $this->textLine('--' . $this->boundary[1]);
        $body .= $this->headerLine('Content-Type', static::CONTENT_TYPE_MULTIPART_RELATED . ':');
        $body .= $this->textLine(' boundary=""' . $this->boundary[2] . '",');
        $body .= $this->textLine(' type=""' . static::CONTENT_TYPE_TEXT_HTML . '"');
        $body .= static::$LE;
        $body .= $this->getBoundary($this->boundary[2], $bodyCharSet, ", $bodyEncoding);
        $body .= $this->encodeString($this->Body, $bodyEncoding);
        $body .= static::$LE;
        $body .= $this->attachAll('inline', $this->boundary[2]);
        $body .= static::$LE;
        $body .= $this->attachAll('attachment', $this->boundary[1]);
        break;
    case 'alt':
        $body .= $mimepre;
        $body .= $this->getBoundary(
            $this->boundary[1],
            $altBodyCharSet,
            static::CONTENT_TYPE_PLAINTEXT,
            $altBodyEncoding

```

```

);
$body .= $this->encodeString($this->AltBody, $altBodyEncoding);
$body .= static::$LE;
$body .= $this->getBoundary(
    $this->boundary[1],
    $bodyCharSet,
    static::CONTENT_TYPE_TEXT_HTML,
    $bodyEncoding
);
$body .= $this->encodeString($this->Body, $bodyEncoding);
$body .= static::$LE;
if (!empty($this->lcal)) {
    $method = static::ICAL_METHOD_REQUEST;
    foreach (static::$lcalMethods as $imethod) {
        if (stripos($this->lcal, 'METHOD:' . $imethod) !== false) {
            $method = $imethod;
            break;
        }
    }
    $body .= $this->getBoundary(
        $this->boundary[1],
        '',
        static::CONTENT_TYPE_TEXT_CALENDAR . ' method=' .
$method,
        ''
    );
    $body .= $this->encodeString($this->lcal, $this->Encoding);
    $body .= static::$LE;
}
$body .= $this->endBoundary($this->boundary[1]);
break;
case 'alt_inline':
$body .= $mimetype;
$body .= $this->getBoundary(
    $this->boundary[1],
    $altBodyCharSet,
    static::CONTENT_TYPE_PLAINTEXT,
    $altBodyEncoding
);
$body .= $this->encodeString($this->AltBody, $altBodyEncoding);
$body .= static::$LE;
$body .= $this->textLine('--' . $this->boundary[1]);

```

```

$body           .= $this->headerLine('Content-Type',
static::CONTENT_TYPE_MULTIPART_RELATED . ':');
$body .= $this->textLine(' boundary="" . $this->boundary[2] . ";"');
$body           .= $this->textLine('           type="" .
static::CONTENT_TYPE_TEXT_HTML . ""');
$body .= static::$LE;
$body .= $this->getBoundary(
    $this->boundary[2],
    $bodyCharSet,
    static::CONTENT_TYPE_TEXT_HTML,
    $bodyEncoding
);
$body .= $this->encodeString($this->Body, $bodyEncoding);
$body .= static::$LE;
$body .= $this->attachAll('inline', $this->boundary[2]);
$body .= static::$LE;
$body .= $this->endBoundary($this->boundary[1]);
break;
case 'alt_attach':
$body .= $mimerep;
$body .= $this->textLine('--' . $this->boundary[1]);
$body           .= $this->headerLine('Content-Type',
static::CONTENT_TYPE_MULTIPART_ALTERNATIVE . ':');
$body .= $this->textLine(' boundary="" . $this->boundary[2] . ""');
$body .= static::$LE;
$body .= $this->getBoundary(
    $this->boundary[2],
    $altBodyCharSet,
    static::CONTENT_TYPE_PLAINTEXT,
    $altBodyEncoding
);
$body .= $this->encodeString($this->AltBody, $altBodyEncoding);
$body .= static::$LE;
$body .= $this->getBoundary(
    $this->boundary[2],
    $bodyCharSet,
    static::CONTENT_TYPE_TEXT_HTML,
    $bodyEncoding
);
$body .= $this->encodeString($this->Body, $bodyEncoding);
$body .= static::$LE;
if (!empty($this->lcal)) {

```

```

$method = static::ICAL_METHOD_REQUEST;
foreach (static::$IcalMethods as $imethod) {
    if (strpos($this->Ical, 'METHOD:' . $imethod) !== false) {
        $method = $imethod;
        break;
    }
}
$body .= $this->getBoundary(
    $this->boundary[2],
    '',
    static::CONTENT_TYPE_TEXT_CALENDAR . ' ; method=' .
$method,
    ''
);
$body .= $this->encodeString($this->Ical, $this->Encoding);
}
$body .= $this->endBoundary($this->boundary[2]);
$body .= static::$LE;
$body .= $this->attachAll('attachment', $this->boundary[1]);
break;
case 'alt_inline_attach':
    $body .= $mimetype;
    $body .= $this->textLine('--' . $this->boundary[1]);
    $body .= $this->headerLine('Content-Type',
static::CONTENT_TYPE_MULTIPART_ALTERNATIVE . ':');
    $body .= $this->textLine(' boundary="" . $this->boundary[2] . ""');
    $body .= static::$LE;
    $body .= $this->getBoundary(
        $this->boundary[2],
        $altBodyCharSet,
        static::CONTENT_TYPE_PLAINTEXT,
        $altBodyEncoding
    );
    $body .= $this->encodeString($this->AltBody, $altBodyEncoding);
    $body .= static::$LE;
    $body .= $this->textLine('--' . $this->boundary[2]);
    $body .= $this->headerLine('Content-Type',
static::CONTENT_TYPE_MULTIPART RELATED . ':');
    $body .= $this->textLine(' boundary="" . $this->boundary[3] . ";"');
    $body .= $this->textLine(' type="" .
static::CONTENT_TYPE_TEXT_HTML . "'");
    $body .= static::$LE;

```

```

$body .= $this->getBoundary(
    $this->boundary[3],
    $bodyCharSet,
    static::CONTENT_TYPE_TEXT_HTML,
    $bodyEncoding
);
$body .= $this->encodeString($this->Body, $bodyEncoding);
$body .= static::$LE;
$body .= $this->attachAll('inline', $this->boundary[3]);
$body .= static::$LE;
$body .= $this->endBoundary($this->boundary[2]);
$body .= static::$LE;
$body .= $this->attachAll('attachment', $this->boundary[1]);
break;
default:
//Catch case 'plain' and case "", applies to simple `text/plain` and
`text/html` body content types
//Reset the `Encoding` property in case we changed it for line length
reasons
$this->Encoding = $bodyEncoding;
$body .= $this->encodeString($this->Body, $this->Encoding);
break;
}

if ($this->isError()) {
    $body = "";
    if ($this->exceptions) {
        throw new Exception($this->lang('empty_message'),
self::STOP_CRITICAL);
    }
} elseif ($this->sign_key_file) {
    try {
        if (!defined('PKCS7_TEXT')) {
            throw new Exception($this->lang('extension_missing') . 'openssl');
        }

        $file = tempnam(sys_get_temp_dir(), 'srcsign');
        $signed = tempnam(sys_get_temp_dir(), 'mailsign');
        file_put_contents($file, $body);

//Workaround for PHP bug https://bugs.php.net/bug.php?id=69197
        if (empty($this->sign_extracerts_file)) {

```

```

$sign = @openssl_pkcs7_sign(
    $file,
    $signed,
    'file://' . realpath($this->sign_cert_file),
    ['file://' . realpath($this->sign_key_file), $this->sign_key_pass],
    []
);
} else {
    $sign = @openssl_pkcs7_sign(
        $file,
        $signed,
        'file://' . realpath($this->sign_cert_file),
        ['file://' . realpath($this->sign_key_file), $this->sign_key_pass],
        [],
        PKCS7_DETACHED,
        $this->sign_extracerts_file
    );
}
}

@unlink($file);
if ($sign) {
    $body = file_get_contents($signed);
    @unlink($signed);
    //The message returned by openssl contains both headers and
    body, so need to split them up
    $parts = explode("\n\n", $body, 2);
    $this->MIMEHeader .= $parts[0] . static::$LE . static::$LE;
    $body = $parts[1];
} else {
    @unlink($signed);
    throw new Exception($this->lang('signing') . openssl_error_string());
}
} catch (Exception $exc) {
    $body = "";
    if ($this->exceptions) {
        throw $exc;
    }
}
}

return $body;
}

```

```

/**
 * Get the boundaries that this message will use
 * @return array
 */
public function getBoundaries()
{
    if (empty($this->boundary)) {
        $this->setBoundaries();
    }
    return $this->boundary;
}

/**
 * Return the start of a message boundary.
 *
 * @param string $boundary
 * @param string $charSet
 * @param string $contentType
 * @param string $encoding
 *
 * @return string
 */
protected function getBoundary($boundary, $charSet, $contentType,
$encoding)
{
    $result = "";
    if (" === $charSet) {
        $charSet = $this->CharSet;
    }
    if (" === $contentType) {
        $contentType = $this->ContentType;
    }
    if (" === $encoding) {
        $encoding = $this->Encoding;
    }
    $result .= $this->textLine('--' . $boundary);
    $result .= sprintf('Content-Type: %s; charset=%s', $contentType,
$charSet);
    $result .= static::$LE;
    //RFC1341 part 5 says 7bit is assumed if not specified
    if (static::ENCODING_7BIT !== $encoding) {

```

```

        $result .= $this->headerLine('Content-Transfer-Encoding', $encoding);
    }
    $result .= static::$LE;

    return $result;
}

/** 
 * Return the end of a message boundary.
 *
 * @param string $boundary
 *
 * @return string
 */
protected function endBoundary($boundary)
{
    return static::$LE . '--' . $boundary . '--' . static::$LE;
}

/** 
 * Set the message type.
 * PHPMailer only supports some preset message types, not arbitrary MIME
structures.
*/
protected function setMessageType()
{
    $type = [];
    if ($this->alternativeExists()) {
        $type[] = 'alt';
    }
    if ($this->inlineImageExists()) {
        $type[] = 'inline';
    }
    if ($this->attachmentExists()) {
        $type[] = 'attach';
    }
    $this->message_type = implode('_', $type);
    if (" === $this->message_type) {
        //The 'plain' message_type refers to the message having a single body
element, not that it is plain-text
        $this->message_type = 'plain';
    }
}

```

```

}

/**
 * Format a header line.
 *
 * @param string $name
 * @param string|int $value
 *
 * @return string
 */
public function headerLine($name, $value)
{
    return $name . ':' . $value . static::$LE;
}

/**
 * Return a formatted mail line.
 *
 * @param string $value
 *
 * @return string
 */
public function textLine($value)
{
    return $value . static::$LE;
}

/**
 * Add an attachment from a path on the filesystem.
 * Never use a user-supplied path to a file!
 * Returns false if the file could not be found or read.
 * Explicitly *does not* support passing URLs; PHPMailer is not an HTTP
client.
 * If you need to do that, fetch the resource yourself and pass it in via a local
file or string.
 *
 * @param string $path      Path to the attachment
 * @param string $name      Overrides the attachment name
 * @param string $encoding  File encoding (see $Encoding)
 * @param string $type       MIME type, e.g. `image/jpeg`; determined
automatically from $path if not specified
 * @param string $disposition Disposition to use

```

```

*
* @throws Exception
*
* @return bool
*/
public function addAttachment(
    $path,
    $name = '',
    $encoding = self::ENCODING_BASE64,
    $type = '',
    $disposition = 'attachment'
) {
    try {
        if (!static::fileIsAccessible($path)) {
            throw new Exception($this->lang('file_access') . $path,
self::STOP_CONTINUE);
        }

        //If a MIME type is not specified, try to work it out from the file name
        if (" === $type) {
            $type = static::filenameToType($path);
        }

        $filename = (string) static::mb_pathinfo($path,
PATHINFO_BASENAME);
        if (" === $name) {
            $name = $filename;
        }
        if (!$this->validateEncoding($encoding)) {
            throw new Exception($this->lang('encoding') . $encoding);
        }

        $this->attachment[] = [
            0 => $path,
            1 => $filename,
            2 => $name,
            3 => $encoding,
            4 => $type,
            5 => false, //isStringAttachment
            6 => $disposition,
            7 => $name,
        ];
    }
}

```

```

} catch (Exception $exc) {
    $this->setError($exc->getMessage());
    $this->edebug($exc->getMessage());
    if ($this->exceptions) {
        throw $exc;
    }

    return false;
}

return true;
}

/**
 * Return the array of attachments.
 *
 * @return array
 */
public function getAttachments()
{
    return $this->attachment;
}

/**
 * Attach all file, string, and binary attachments to the message.
 * Returns an empty string on failure.
 *
 * @param string $disposition_type
 * @param string $boundary
 *
 * @throws Exception
 *
 * @return string
 */
protected function attachAll($disposition_type, $boundary)
{
    //Return text of body
    $mime = [];
    $cidUniq = [];
    $incl = [];

    //Add all attachments

```

```

foreach ($this->attachment as $attachment) {
    //Check if it is a valid disposition_filter
    if ($attachment[6] === $disposition_type) {
        //Check for string attachment
        $string = "";
        $path = "";
        $bString = $attachment[5];
        if ($bString) {
            $string = $attachment[0];
        } else {
            $path = $attachment[0];
        }

        $inlhash = hash('sha256', serialize($attachment));
        if (in_array($inlhash, $inl, true)) {
            continue;
        }
        $inl[] = $inlhash;
        $name = $attachment[2];
        $encoding = $attachment[3];
        $type = $attachment[4];
        $disposition = $attachment[6];
        $cid = $attachment[7];
        if ('inline' === $disposition && array_key_exists($cid, $cidUniq)) {
            continue;
        }
        $cidUniq[$cid] = true;

        $mime[] = sprintf('--%s%s', $boundary, static::$LE);
        //Only include a filename property if we have one
        if (!empty($name)) {
            $mime[] = sprintf(
                'Content-Type: %s; name=%s%s',
                $type,
                static::quotedString($this->encodeHeader($this-
>secureHeader($name))),
                static::$LE
            );
        } else {
            $mime[] = sprintf(
                'Content-Type: %s%s',
                $type,

```

```

        static::$LE
    );
}

//RFC1341 part 5 says 7bit is assumed if not specified
if (static::ENCODING_7BIT !== $encoding) {
    $mime[] = sprintf('Content-Transfer-Encoding: %s%s', $encoding,
static::$LE);
}

//Only set Content-IDs on inline attachments
if ((string) $cid !== " && $disposition === 'inline') {
    $mime[] = 'Content-ID: <' . $this->encodeHeader($this-
>secureHeader($cid)) . '>' . static::$LE;
}

//Allow for bypassing the Content-Disposition header
if (!empty($disposition)) {
    $encoded_name = $this->encodeHeader($this-
>secureHeader($name));
    if (!empty($encoded_name)) {
        $mime[] = sprintf(
            'Content-Disposition: %s; filename=%s%s',
            $disposition,
            static::quotedString($encoded_name),
            static::$LE . static::$LE
        );
    } else {
        $mime[] = sprintf(
            'Content-Disposition: %s%s',
            $disposition,
            static::$LE . static::$LE
        );
    }
} else {
    $mime[] = static::$LE;
}

//Encode as string attachment
if ($bString) {
    $mime[] = $this->encodeString($string, $encoding);
} else {
    $mime[] = $this->encodeFile($path, $encoding);
}

```

```

        }
        if ($this->isError()) {
            return "";
        }
        $mime[] = static::$LE;
    }
}

$mime[] = sprintf('--%s--%s', $boundary, static::$LE);

return implode("\n", $mime);
}

/** 
 * Encode a file attachment in requested format.
 * Returns an empty string on failure.
 *
 * @param string $path    The full path to the file
 * @param string $encoding The encoding to use; one of 'base64', '7bit', '8bit',
 * 'binary', 'quoted-printable'
 *
 * @return string
 */
protected function encodeFile($path, $encoding = self::ENCODING_BASE64)
{
    try {
        if (!static::fileIsAccessible($path)) {
            throw new Exception($this->lang('file_open') . $path,
self::STOP_CONTINUE);
        }
        $file_buffer = file_get_contents($path);
        if (false === $file_buffer) {
            throw new Exception($this->lang('file_open') . $path,
self::STOP_CONTINUE);
        }
        $file_buffer = $this->encodeString($file_buffer, $encoding);

        return $file_buffer;
    } catch (Exception $exc) {
        $this->setError($exc->getMessage());
        $this->edebug($exc->getMessage());
        if ($this->exceptions) {

```

```

        throw $exc;
    }

    return "";
}

/** 
 * Encode a string in requested format.
 * Returns an empty string on failure.
 *
 * @param string $str      The text to encode
 * @param string $encoding The encoding to use; one of 'base64', '7bit', '8bit',
 * 'binary', 'quoted-printable'
 *
 * @throws Exception
 *
 * @return string
 */
public function encodeString($str, $encoding = self::ENCODING_BASE64)
{
    $encoded = "";
    switch (strtolower($encoding)) {
        case static::ENCODING_BASE64:
            $encoded = chunk_split(
                base64_encode($str),
                static::STD_LINE_LENGTH,
                static::$LE
            );
            break;
        case static::ENCODING_7BIT:
        case static::ENCODING_8BIT:
            $encoded = static::normalizeBreaks($str);
            //Make sure it ends with a line break
            if (substr($encoded, -(strlen(static::$LE))) !== static::$LE) {
                $encoded .= static::$LE;
            }
            break;
        case static::ENCODING_BINARY:
            $encoded = $str;
            break;
        case static::ENCODING_QUOTED_PRINTABLE:

```

```

$encoded = $this->encodeQP($str);
break;
default:
    $this->setError($this->lang('encoding') . $encoding);
    if ($this->exceptions) {
        throw new Exception($this->lang('encoding') . $encoding);
    }
    break;
}

return $encoded;
}

/** 
 * Encode a header value (not including its label) optimally.
 * Picks shortest of Q, B, or none. Result includes folding if needed.
 * See RFC822 definitions for phrase, comment and text positions,
 * and RFC2047 for inline encodings.
 *
 * @param string $str      The header value to encode
 * @param string $position What context the string will be used in
 *
 * @return string
 */
public function encodeHeader($str, $position = 'text')
{
    $position = strtolower($position);
    if ($this->UseSMTPUTF8 && !("comment" === $position)) {
        return trim(static::normalizeBreaks($str));
    }

    $matchcount = 0;
    switch (strtolower($position)) {
        case 'phrase':
            if (!preg_match('/[\\200-\\377]/', $str)) {
                //Can't use addslashes as we don't know the value of
                magic_quotes_sybase
                $encoded = addcslashes($str, "\0..\37\177\\\"");
                if   (($str === $encoded) && !preg_match('/[^A-Za-z0-
9!#$%&\'*+\\=?^`{|}~ -]/', $str)) {
                    return $encoded;
                }
            }
    }
}

```

```

        return "\"$encoded\"";
    }
    $matchcount = preg_match_all('/[^\\040\\041\\043-\\133\\135-\\176]/', $str,
$matches);
    break;
/* @noinspection PhpMissingBreakStatementInspection */
case 'comment':
    $matchcount = preg_match_all('/[()]/', $str, $matches);
//fallthrough
case 'text':
default:
    $matchcount += preg_match_all('/[\000-\010\013\014\016-\037\177-\1377]/', $str, $matches);
    break;
}

if ($this->has8bitChars($str)) {
    $charset = $this->CharSet;
} else {
    $charset = static::CHARSET_ASCII;
}

//Q/B encoding adds 8 chars and the charset ("=?<charset>?[QB]?<content>?=").  

$overhead = 8 + strlen($charset);

if ('mail' === $this->Mailer) {
    $maxlen = static::MAIL_MAX_LINE_LENGTH - $overhead;
} else {
    $maxlen = static::MAX_LINE_LENGTH - $overhead;
}

//Select the encoding that produces the shortest output and/or prevents
corruption.
if ($matchcount > strlen($str) / 3) {
    //More than 1/3 of the content needs encoding, use B-encode.
    $encoding = 'B';
} elseif ($matchcount > 0) {
    //Less than 1/3 of the content needs encoding, use Q-encode.
    $encoding = 'Q';
} elseif (strlen($str) > $maxlen) {

```

```

//No encoding needed, but value exceeds max line length, use Q-encode
to prevent corruption.
$encoding = 'Q';
} else {
    //No reformatting needed
    $encoding = false;
}

switch ($encoding) {
    case 'B':
        if ($this->hasMultiBytes($str)) {
            //Use a custom function which correctly encodes and wraps long
            //multibyte strings without breaking lines within a character
            $encoded = $this->base64EncodeWrapMB($str, "\n");
        } else {
            $encoded = base64_encode($str);
            $maxlen -= $maxlen % 4;
            $encoded = trim(chunk_split($encoded, $maxlen, "\n"));
        }
        $encoded = preg_replace('/^(.*$)/m', '=?' . $charset . .
"?$encoding?\1?=", $encoded);
        break;
    case 'Q':
        $encoded = $this->encodeQ($str, $position);
        $encoded = $this->wrapText($encoded, $maxlen, true);
        $encoded = str_replace('='.$static::LE, "\n", trim($encoded));
        $encoded = preg_replace('/^(.*$)/m', '=?' . $charset . .
"?$encoding?\1?=", $encoded);
        break;
    default:
        return $str;
}

return trim(static::normalizeBreaks($encoded));
}

/**
 * Check if a string contains multi-byte characters.
 *
 * @param string $str multi-byte text to wrap encode
 *
 * @return bool

```

```

*/
public function hasMultiBytes($str)
{
    if (function_exists('mb_strlen')) {
        return strlen($str) > mb_strlen($str, $this->CharSet);
    }

    //Assume no multibytes (we can't handle without mbstring functions
    //anyway)
    return false;
}

/**
 * Does a string contain any 8-bit chars (in any charset)?
 *
 * @param string $text
 *
 * @return bool
 */
public function has8bitChars($text)
{
    return (bool) preg_match('/[\x80-\xFF]/', $text);
}

/**
 * Encode and wrap long multibyte strings for mail headers
 * without breaking lines within a character.
 * Adapted from a function by paravoid.
 *
 * @see https://www.php.net/manual/en/function.mb-encode-
mimeheader.php#60283
 *
 * @param string $str    multi-byte text to wrap encode
 * @param string $linebreak string to use as linefeed/end-of-line
 *
 * @return string
 */
public function base64EncodeWrapMB($str, $linebreak = null)
{
    $start = '=?' . $this->CharSet . '?B?';
    $end = '?=';
    $encoded = '';

```

```

if (null === $linebreak) {
    $linebreak = static::$LE;
}

$mb_length = mb_strlen($str, $this->CharSet);
//Each line must have length <= 75, including $start and $end
[length] = 75 - strlen($start) - strlen($end);
//Average multi-byte ratio
$ratio = $mb_length / strlen($str);
//Base64 has a 4:3 ratio
$avgLength = floor($length * $ratio * .75);

$offset = 0;
for ($i = 0; $i < $mb_length; $i += $offset) {
    $lookBack = 0;
    do {
        $offset = $avgLength - $lookBack;
        $chunk = mb_substr($str, $i, $offset, $this->CharSet);
        $chunk = base64_encode($chunk);
        ++$lookBack;
    } while (strlen($chunk) > $length);
    $encoded .= $chunk . $linebreak;
}

//Chomp the last linefeed
return substr($encoded, 0, -strlen($linebreak));
}

/** 
 * Encode a string in quoted-printable format.
 * According to RFC2045 section 6.7.
 *
 * @param string $string The text to encode
 *
 * @return string
 */
public function encodeQP($string)
{
    return static::normalizeBreaks(quoted_printable_encode($string));
}

/**

```

```

* Encode a string using Q encoding.
*
* @see https://www.rfc-editor.org/rfc/rfc2047#section-4.2
*
* @param string $str    the text to encode
* @param string $position Where the text is going to be used, see the RFC
for what that means
*
* @return string
*/
public function encodeQ($str, $position = 'text')
{
    //There should not be any EOL in the string
    $pattern = "";
    $encoded = str_replace(["\r", "\n"], "", $str);
    switch (strtolower($position)) {
        case 'phrase':
            //RFC 2047 section 5.3
            $pattern = '^A-Za-z0-9!*+\/ -';
            break;
        /*
         * RFC 2047 section 5.2.
         * Build $pattern without including delimiters and []
         */
        /* @noinspection PhpMissingBreakStatementInspection */
        case 'comment':
            $pattern = '\(\)';
            /* Intentional fall through */
        case 'text':
        default:
            //RFC 2047 section 5.1
            //Replace every high ascii, control, =, ? and _ characters
            $pattern = '\000-\011\013\014\016-\037\075\077\137\177-\377' . $pattern;
            break;
    }
    $matches = [];
    if (preg_match_all("/[$pattern]/", $encoded, $matches)) {
        //If the string contains an '=', make sure it's the first thing we replace
        //so as to avoid double-encoding
        $eqkey = array_search('=', $matches[0], true);
        if (false !== $eqkey) {

```

```

        unset($matches[0][$eqkey]);
        array_unshift($matches[0], '=');
    }
    foreach (array_unique($matches[0]) as $char) {
        $encoded = str_replace($char, '=' . sprintf('%02X', ord($char)),
$encoded);
    }
}

//Replace spaces with _ (more readable than =20)
//RFC 2047 section 4.2(2)
return str_replace(' ', '_', $encoded);
}

/**
 * Add a string or binary attachment (non-filesystem).
 * This method can be used to attach ascii or binary data,
 * such as a BLOB record from a database.
 *
 * @param string $string      String attachment data
 * @param string $filename    Name of the attachment
 * @param string $encoding    File encoding (see $Encoding)
 * @param string $type        File extension (MIME) type
 * @param string $disposition Disposition to use
 *
 * @throws Exception
 *
 * @return bool True on successfully adding an attachment
 */
public function addStringAttachment(
    $string,
    $filename,
    $encoding = self::ENCODING_BASE64,
    $type = '',
    $disposition = 'attachment'
) {
    try {
        //If a MIME type is not specified, try to work it out from the file name
        if (" === $type) {
            $type = static::filenameToType($filename);
        }

        if (!$this->validateEncoding($encoding)) {

```

```

        throw new Exception($this->lang('encoding') . $encoding);
    }

    //Append to $attachment array
    $this->attachment[] = [
        0 => $string,
        1 => $filename,
        2 => static::mb_pathinfo($filename, PATHINFO_BASENAME),
        3 => $encoding,
        4 => $type,
        5 => true, //isStringAttachment
        6 => $disposition,
        7 => 0,
    ];
} catch (Exception $exc) {
    $this->setError($exc->getMessage());
    $this->edebug($exc->getMessage());
    if ($this->exceptions) {
        throw $exc;
    }
}

return false;
}

return true;
}

/**
 * Add an embedded (inline) attachment from a file.
 * This can include images, sounds, and just about any other document type.
 * These differ from 'regular' attachments in that they are intended to be
 * displayed inline with the message, not just attached for download.
 * This is used in HTML messages that embed the images
 * the HTML refers to using the '$cid' value in `img` tags, for example ``.
 * Never use a user-supplied path to a file!
 *
 * @param string $path      Path to the attachment
 * @param string $cid      Content ID of the attachment; Use this to reference
 *                        the content when using an embedded image in HTML
 * @param string $name     Overrides the attachment filename

```

```

    * @param string $encoding      File encoding (see $Encoding) defaults to
`base64`
    * @param string $type          File MIME type (by default mapped from the
`$path` filename's extension)
    * @param string $disposition  Disposition to use: `inline` (default) or
`attachment`
    *
        (unlikely you want this – {@see `addAttachment()`} instead)
    *

    * @return bool True on successfully adding an attachment
    * @throws Exception
    *
    */
public function addEmbeddedImage(
    $path,
    $cid,
    $name = '',
    $encoding = self::ENCODING_BASE64,
    $type = '',
    $disposition = 'inline'
) {
    try {
        if (!static::fileIsAccessible($path)) {
            throw new Exception($this->lang('file_access') . $path,
self::STOP_CONTINUE);
        }

        //If a MIME type is not specified, try to work it out from the file name
        if (" === $type) {
            $type = static::filenameToType($path);
        }

        if (!$this->validateEncoding($encoding)) {
            throw new Exception($this->lang('encoding') . $encoding);
        }

        $filename      =      (string)      static::mb_pathinfo($path,
PATHINFO_BASENAME);
        if (" === $name) {
            $name = $filename;
        }

        //Append to $attachment array
    }
}

```

```

$this->attachment[] = [
    0 => $path,
    1 => $filename,
    2 => $name,
    3 => $encoding,
    4 => $type,
    5 => false, //isStringAttachment
    6 => $disposition,
    7 => $cid,
];
} catch (Exception $exc) {
    $this->setError($exc->getMessage());
    $this->edebug($exc->getMessage());
    if ($this->exceptions) {
        throw $exc;
    }
}

return false;
}

return true;
}

/**
 * Add an embedded stringified attachment.
 * This can include images, sounds, and just about any other document type.
 * If your filename doesn't contain an extension, be sure to set the $type to an
appropriate MIME type.
 *
 * @param string $string      The attachment binary data
 * @param string $cid         Content ID of the attachment; Use this to reference
 *                           the content when using an embedded image in HTML
 * @param string $name        A filename for the attachment. If this contains an
extension,
 *                           PHPMailer will attempt to set a MIME type for the attachment.
 *                           For example 'file.jpg' would get an 'image/jpeg' MIME type.
 * @param string $encoding    File encoding (see $Encoding), defaults to
'base64'
 * @param string $type        MIME type - will be used in preference to any
automatically derived type
 * @param string $disposition Disposition to use
 *

```

```

* @throws Exception
*
* @return bool True on successfully adding an attachment
*/
public function addStringEmbeddedImage(
    $string,
    $cid,
    $name = '',
    $encoding = self::ENCODING_BASE64,
    $type = '',
    $disposition = 'inline'
) {
    try {
        //If a MIME type is not specified, try to work it out from the name
        if (" === $type && !empty($name)) {
            $type = static::filenameToType($name);
        }

        if (!$this->validateEncoding($encoding)) {
            throw new Exception($this->lang('encoding') . $encoding);
        }

        //Append to $attachment array
        $this->attachment[] = [
            0 => $string,
            1 => $name,
            2 => $name,
            3 => $encoding,
            4 => $type,
            5 => true, //isStringAttachment
            6 => $disposition,
            7 => $cid,
        ];
    } catch (Exception $exc) {
        $this->setError($exc->getMessage());
        $this->edebug($exc->getMessage());
        if ($this->exceptions) {
            throw $exc;
        }
    }

    return false;
}

```

```

        return true;
    }

/***
 * Validate encodings.
 *
 * @param string $encoding
 *
 * @return bool
 */
protected function validateEncoding($encoding)
{
    return in_array(
        $encoding,
        [
            self::ENCODING_7BIT,
            self::ENCODING_QUOTED_PRINTABLE,
            self::ENCODING_BASE64,
            self::ENCODING_8BIT,
            self::ENCODING_BINARY,
        ],
        true
    );
}

/***
 * Check if an embedded attachment is present with this cid.
 *
 * @param string $cid
 *
 * @return bool
 */
protected function cidExists($cid)
{
    foreach ($this->attachment as $attachment) {
        if ('inline' === $attachment[6] && $cid === $attachment[7]) {
            return true;
        }
    }
}

return false;

```

```

}

/** 
 * Check if an inline attachment is present.
 *
 * @return bool
 */
public function inlineImageExists()
{
    foreach ($this->attachment as $attachment) {
        if ('inline' === $attachment[6]) {
            return true;
        }
    }

    return false;
}

/** 
 * Check if an attachment (non-inline) is present.
 *
 * @return bool
 */
public function attachmentExists()
{
    foreach ($this->attachment as $attachment) {
        if ('attachment' === $attachment[6]) {
            return true;
        }
    }

    return false;
}

/** 
 * Check if this message has an alternative body set.
 *
 * @return bool
 */
public function alternativeExists()
{
    return !empty($this->AltBody);
}

```

```

}

/**
 * Clear queued addresses of given kind.
 *
 * @param string $kind 'to', 'cc', or 'bcc'
 */
public function clearQueuedAddresses($kind)
{
    $this->RecipientsQueue = array_filter(
        $this->RecipientsQueue,
        static function ($params) use ($kind) {
            return $params[0] !== $kind;
        }
    );
}

/**
 * Clear all To recipients.
 */
public function clearAddresses()
{
    foreach ($this->to as $to) {
        unset($this->all_recipients[strtolower($to[0])]);
    }
    $this->to = [];
    $this->clearQueuedAddresses('to');
}

/**
 * Clear all CC recipients.
 */
public function clearCCs()
{
    foreach ($this->cc as $cc) {
        unset($this->all_recipients[strtolower($cc[0])]);
    }
    $this->cc = [];
    $this->clearQueuedAddresses('cc');
}

/**

```

```

* Clear all BCC recipients.
*/
public function clearBCCs()
{
    foreach ($this->bcc as $bcc) {
        unset($this->all_recipients[strtolower($bcc[0])]);
    }
    $this->bcc = [];
    $this->clearQueuedAddresses('bcc');
}

/** 
 * Clear all ReplyTo recipients.
*/
public function clearReplyTos()
{
    $this->ReplyTo = [];
    $this->ReplyToQueue = [];
}

/** 
 * Clear all recipient types.
*/
public function clearAllRecipients()
{
    $this->to = [];
    $this->cc = [];
    $this->bcc = [];
    $this->all_recipients = [];
    $this->RecipientsQueue = [];
}

/** 
 * Clear all filesystem, string, and binary attachments.
*/
public function clearAttachments()
{
    $this->attachment = [];
}

/** 
 * Clear all custom headers.

```

```

*/
public function clearCustomHeaders()
{
    $this->CustomHeader = [];
}

/** 
 * Clear a specific custom header by name or name and value.
 * $name value can be overloaded to contain
 * both header name and value (name:value).
 *
 * @param string    $name  Custom header name
 * @param string|null $value Header value
 *
 * @return bool True if a header was replaced successfully
 */
public function clearCustomHeader($name, $value = null)
{
    if (null === $value && strpos($name, ':') !== false) {
        //Value passed in as name:value
        list($name, $value) = explode(':', $name, 2);
    }
    $name = trim($name);
    $value = (null === $value) ? null : trim($value);

    foreach ($this->CustomHeader as $k => $pair) {
        if ($pair[0] == $name) {
            // We remove the header if the value is not provided or it matches.
            if (null === $value || $pair[1] == $value) {
                unset($this->CustomHeader[$k]);
            }
        }
    }

    return true;
}

/** 
 * Replace a custom header.
 * $name value can be overloaded to contain
 * both header name and value (name:value).
 *

```

```

* @param string    $name  Custom header name
* @param string|null $value Header value
*
* @return bool True if a header was replaced successfully
* @throws Exception
*/
public function replaceCustomHeader($name, $value = null)
{
    if (null === $value && strpos($name, ':') !== false) {
        //Value passed in as name:value
        list($name, $value) = explode(':', $name, 2);
    }
    $name = trim($name);
    $value = (null === $value) ? " : trim($value);

    $replaced = false;
    foreach ($this->CustomHeader as $k => $pair) {
        if ($pair[0] == $name) {
            if ($replaced) {
                unset($this->CustomHeader[$k]);
                continue;
            }
            if (strpos($name . $value, "\r\n") !== false) {
                if ($this->exceptions) {
                    throw new Exception($this->lang('invalid_header'));
                }
            }
            return false;
        }
        $this->CustomHeader[$k] = [$name, $value];
        $replaced = true;
    }
}

return true;
}

/**
 * Add an error message to the error container.
 *
 * @param string $msg
 */

```

```

protected function setError($msg)
{
    ++$this->error_count;
    if ('smtp' === $this->Mailer && null !== $this->smtp) {
        $lasterror = $this->smtp->getError();
        if (!empty($lasterror['error'])) {
            $msg .= '' . $this->lang('smtp_error') . $lasterror['error'];
            if (!empty($lasterror['detail'])) {
                $msg .= '' . $this->lang('smtp_detail') . $lasterror['detail'];
            }
            if (!empty($lasterror['smtp_code'])) {
                $msg .= ' ' . $this->lang('smtp_code') . $lasterror['smtp_code'];
            }
            if (!empty($lasterror['smtp_code_ex'])) {
                $msg .= ' ' . $this->lang('smtp_code_ex');
                $lasterror['smtp_code_ex'];
            }
        }
        $this->ErrorInfo = $msg;
    }

    /**
     * Return an RFC 822 formatted date.
     *
     * @return string
     */
    public static function rfcDate()
    {
        //Set the time zone to whatever the default is to avoid 500 errors
        //Will default to UTC if it's not set properly in php.ini
        date_default_timezone_set(@date_default_timezone_get());

        return date('D, j M Y H:i:s O');
    }

    /**
     * Get the server hostname.
     * Returns 'localhost.localdomain' if unknown.
     *
     * @return string
     */

```

```

protected function serverHostname()
{
    $result = "";
    if (!empty($this->Hostname)) {
        $result = $this->Hostname;
    } elseif (isset($_SERVER) && array_key_exists('SERVER_NAME',
$_SERVER)) {
        $result = $_SERVER['SERVER_NAME'];
    } elseif (function_exists('gethostname') && gethostname() !== false) {
        $result = gethostname();
    } elseif (php_uname('n') !== "") {
        $result = php_uname('n');
    }
    if (!static::isValidHost($result)) {
        return 'localhost.localdomain';
    }

    return $result;
}

/**
 * Validate whether a string contains a valid value to use as a hostname or IP
 * address.
 * IPv6 addresses must include [], e.g. `[:1]`, not just `:1`.
 *
 * @param string $host The host name or IP address to check
 *
 * @return bool
 */
public static function isValidHost($host)
{
    //Simple syntax limits
    if (
        empty($host)
        || !is_string($host)
        || strlen($host) > 256
        || !preg_match('/^([a-z\d.-]*|[[a-f\d:]+\])$/i', $host)
    ) {
        return false;
    }
    //Looks like a bracketed IPv6 address
}

```

```

    if (strlen($host) > 2 && substr($host, 0, 1) === '[' && substr($host, -1, 1) === ']') {
        return filter_var(substr($host, 1, -1), FILTER_VALIDATE_IP,
FILTER_FLAG_IPV6) !== false;
    }
    //If removing all the dots results in a numeric string, it must be an IPv4
address.
    //Need to check this first because otherwise things like `999.0.0.0` are
considered valid host names
    if (is_numeric(str_replace('.', '', $host))) {
        //Is it a valid IPv4 address?
        return filter_var($host, FILTER_VALIDATE_IP, FILTER_FLAG_IPV4)
!== false;
    }
    //Is it a syntactically valid hostname (when embedded in a URL)?
    return filter_var('https://' . $host, FILTER_VALIDATE_URL) !== false;
}

/**
 * Check whether the supplied address uses Unicode in the local part.
 *
 * @return bool
 */
protected function addressHasUnicodeLocalPart($address)
{
    return (bool) preg_match('/[\x80-\xFF].*@/', $address);
}

/**
 * Check whether any of the supplied addresses use Unicode in the local part.
 *
 * @return bool
 */
protected function anyAddressHasUnicodeLocalPart($addresses)
{
    foreach ($addresses as $address) {
        if (is_array($address)) {
            $address = $address[0];
        }
        if ($this->addressHasUnicodeLocalPart($address)) {
            return true;
        }
    }
}

```

```

        }
        return false;
    }

    /**
     * Check whether the message requires SMTPUTF8 based on what's known
     * so far.
     *
     * @return bool
     */
    public function needsSMTPUTF8()
    {
        return $this->UseSMTPUTF8;
    }

    /**
     * Get an error message in the current language.
     *
     * @param string $key
     *
     * @return string
     */
    protected function lang($key)
    {
        if (count($this->language) < 1) {
            $this->setLanguage(); //Set the default language
        }

        if (array_key_exists($key, $this->language)) {
            if ('smtp_connect_failed' === $key) {
                //Include a link to troubleshooting docs on SMTP connection failure.
                //This is by far the biggest cause of support questions
                //but it's usually not PHPMailer's fault.
                return $this->language[$key]
                . '  

https://github.com/PHPMailer/PHPMailer/wiki/Troubleshooting';
            }
        }

        return $this->language[$key];
    }

    //Return the key as a fallback

```

```

        return $key;
    }

    /**
     * Build an error message starting with a generic one and adding details if
possible.
     *
     * @param string $base_key
     * @return string
     */
private function getSmtpErrorMessage($base_key)
{
    $message = $this->lang($base_key);
    $error = $this->smtp->getError();
    if (!empty($error['error'])) {
        $message .= '' . $error['error'];
        if (!empty($error['detail'])) {
            $message .= '' . $error['detail'];
        }
    }
}

return $message;
}

/**
 * Check if an error occurred.
 *
 * @return bool True if an error did occur
 */
public function isError()
{
    return $this->error_count > 0;
}

/**
 * Add a custom header.
 * $name value can be overloaded to contain
 * both header name and value (name:value).
 *
 * @param string    $name Custom header name
 * @param string|null $value Header value
 *

```

```

* @return bool True if a header was set successfully
* @throws Exception
*/
public function addCustomHeader($name, $value = null)
{
    if (null === $value && strpos($name, ':') !== false) {
        //Value passed in as name:value
        list($name, $value) = explode(':', $name, 2);
    }
    $name = trim($name);
    $value = (null === $value) ? " : trim($value);
    //Ensure name is not empty, and that neither name nor value contain line
    breaks
    if (empty($name) || strpbrk($name . $value, "\r\n") !== false) {
        if ($this->exceptions) {
            throw new Exception($this->lang('invalid_header'));
        }
    }

    return false;
}
$this->CustomHeader[] = [$name, $value];

    return true;
}

/**
 * Returns all custom headers.
 *
 * @return array
 */
public function getCustomHeaders()
{
    return $this->CustomHeader;
}

/**
 * Create a message body from an HTML string.
 * Automatically inlines images and creates a plain-text version by converting
the HTML,
    * overwriting any existing values in Body and AltBody.
    * Do not source $message content from user input!

```

```

    * $basedir is prepended when handling relative URLs, e.g.  and must not be empty
    * will look for an image file in $basedir/images/a.png and convert it to inline.
    * If you don't provide a $basedir, relative paths will be left untouched (and thus
probably break in email)
    * Converts data-uri images into embedded attachments.
    * If you don't want to apply these transformations to your HTML, just set Body
and AltBody directly.
    *
    * @param string      $message  HTML message string
    * @param string      $basedir Absolute path to a base directory to prepend
to relative paths to images
    * @param bool|callable $advanced Whether to use the internal HTML to text
converter
    *                      or your own custom converter
    * @return string The transformed message body
    *
    * @throws Exception
    *
    * @see PHPMailer::html2text()
*/

```

public function msgHTML(\$message, \$basedir = "", \$advanced = false)

```

{
    preg_match_all('/(?!<!--)(src|background)=["\'](.*)["\']/Ui',           $message,
$images);
    if (array_key_exists(2, $images)) {
        if (strlen($basedir) > 1 && '/' !== substr($basedir, -1)) {
            //Ensure $basedir has a trailing /
            $basedir .= '/';
        }
        foreach ($images[2] as $imgindex => $url) {
            //Convert data URIs into embedded images
            //e.g.
"data:image/gif;base64,R0lGODlhAQABAAAAACH5BAEKAEEALAAAAAABAA
EAAAICTAEAOw=="
            $match = [];
            if (preg_match('#^data:(image/(?:jpe?g|gif|png));?(base64)?,(.+)#',
$url, $match)) {
                if (count($match) === 4 && static::ENCODING_BASE64 ===
$match[2]) {
                    $data = base64_decode($match[3]);
                } elseif (" === $match[2]) {

```

```

        $data = rawurldecode($match[3]);
    } else {
        //Not recognised so leave it alone
        continue;
    }
    //Hash the decoded data, not the URL, so that the same data-URI
image used in multiple places
    //will only be embedded once, even if it used a different encoding
    $cid = substr(hash('sha256', $data), 0, 32) . '@phpmailer.0';
//RFC2392 S 2

if (!$this->cidExists($cid)) {
    $this->addStringEmbeddedImage(
        $data,
        $cid,
        'embed' . $imgindex,
        static::ENCODING_BASE64,
        $match[1]
    );
}
$message = str_replace(
    $images[0][$imgindex],
    $images[1][$imgindex] . '="cid:' . $cid . '"',
    $message
);
continue;
}
if (
    //Only process relative URLs if a basedir is provided (i.e. no absolute
local paths)
    !empty($basedir)
    //Ignore URLs containing parent dir traversal (..)
    && (strpos($url, '..') === false)
    //Do not change urls that are already inline images
    && 0 !== strpos($url, 'cid:')
    //Do not change absolute URLs, including anonymous protocol
    && !preg_match('#^([a-z][a-z0-9+-]*://[^#]*)$', $url)
) {
    $filename = static::mb_pathinfo($url, PATHINFO_BASENAME);
    $directory = dirname($url);
    if ('.' === $directory) {
        $directory = '';

```

```

}

//RFC2392 S 2
$cid = substr(hash('sha256', $url), 0, 32) . '@phpmailer.0';
if (strlen($basedir) > 1 && '/' !== substr($basedir, -1)) {
    $basedir .= '/';
}
if (strlen($directory) > 1 && '/' !== substr($directory, -1)) {
    $directory .= '/';
}
if (
    $this->addEmbeddedImage(
        $basedir . $directory . $filename,
        $cid,
        $filename,
        static::ENCODING_BASE64,
        static::_mime_types((string) static::mb_pathinfo($filename,
PATHINFO_EXTENSION)))
)
){
    $message = preg_replace(
        '/' . $images[1][$imgindex] . '=["\']' . preg_quote($url, '/') .
        '['"\']/Ui',
        $images[1][$imgindex] . '="cid:' . $cid . """",
        $message
    );
}
}
}
}

$this->isHTML();
//Convert all message body line breaks to LE, makes quoted-printable
encoding work much better
$this->Body = static::normalizeBreaks($message);
$this->AltBody = static::normalizeBreaks($this->html2text($message,
$advanced));
if (!$this->alternativeExists()) {
    $this->AltBody = 'This is an HTML-only message. To view it, activate
HTML in your email application.'
    . static::$LE;
}

return $this->Body;

```

```

}

/** 
 * Convert an HTML string into plain text.
 * This is used by msgHTML().
 * Note - older versions of this function used a bundled advanced converter
 * which was removed for license reasons in #232.
 * Example usage:
 *
 * ````php
 * //Use default conversion
 * $plain = $mail->html2text($html);
 * //Use your own custom converter
 * $plain = $mail->html2text($html, function($html) {
 *     $converter = new MyHtml2text($html);
 *     return $converter->get_text();
 * });
 * ```
 *
 * @param string      $html    The HTML text to convert
 * @param bool|callable $advanced Any boolean value to use the internal
converter,
 *
 *                      or provide your own callable for custom conversion.
 *
 *                      *Never* pass user-supplied data into this parameter
 *
 * @return string
 */
public function html2text($html, $advanced = false)
{
    if (is_callable($advanced)) {
        return call_user_func($advanced, $html);
    }

    return html_entity_decode(
        trim(strip_tags(preg_replace('/<(head|title|style|script)[^>]*>.*?<\\\1>/si',
", $html))),
        ENT_QUOTES,
        $this->CharSet
    );
}

/**
```

```

* Get the MIME type for a file extension.
*
* @param string $ext File extension
*
* @return string MIME type of file
*/
public static function _mime_types($ext = "")
{
    $mimes = [
        'xl' => 'application/excel',
        'js' => 'application/javascript',
        'hqx' => 'application/mac-binhex40',
        'cpt' => 'application/mac-compactpro',
        'bin' => 'application/macbinary',
        'doc' => 'application/msword',
        'word' => 'application/msword',
        'xlsx' => 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet',
        'xltx' => 'application/vnd.openxmlformats-officedocument.spreadsheetml.template',
        'potx' => 'application/vnd.openxmlformats-officedocument.presentationml.template',
        'ppsx' => 'application/vnd.openxmlformats-officedocument.presentationml.slideshow',
        'pptx' => 'application/vnd.openxmlformats-officedocument.presentationml.presentation',
        'sldx' => 'application/vnd.openxmlformats-officedocument.presentationml.slide',
        'docx' => 'application/vnd.openxmlformats-officedocument.wordprocessingml.document',
        'dotx' => 'application/vnd.openxmlformats-officedocument.wordprocessingml.template',
        'xlam' => 'application/vnd.ms-excel.addin.macroEnabled.12',
        'xlsb' => 'application/vnd.ms-excel.sheet.binary.macroEnabled.12',
        'class' => 'application/octet-stream',
        'dll' => 'application/octet-stream',
        'dms' => 'application/octet-stream',
        'exe' => 'application/octet-stream',
        'lha' => 'application/octet-stream',
        'lzh' => 'application/octet-stream',
        'psd' => 'application/octet-stream',
        'sea' => 'application/octet-stream',
    ];
}

```

```
'so' => 'application/octet-stream',
'oda' => 'application/oda',
'pdf' => 'application/pdf',
'ai' => 'application/postscript',
'eps' => 'application/postscript',
'ps' => 'application/postscript',
'smi' => 'application/smil',
'smil' => 'application/smil',
'mif' => 'application/vnd.mif',
'xls' => 'application/vnd.ms-excel',
'ppt' => 'application/vnd.ms-powerpoint',
'wbxml' => 'application/vnd.wap.wbxml',
'wmlc' => 'application/vnd.wap.wmlc',
'dcr' => 'application/x-director',
'dir' => 'application/x-director',
'dxr' => 'application/x-director',
'dvi' => 'application/x-dvi',
'gtar' => 'application/x-gtar',
'php3' => 'application/x-httpd-php',
'php4' => 'application/x-httpd-php',
'php' => 'application/x-httpd-php',
'phtml' => 'application/x-httpd-php',
'phps' => 'application/x-httpd-php-source',
'swf' => 'application/x-shockwave-flash',
'sit' => 'application/x-stuffit',
'tar' => 'application/x-tar',
'tgz' => 'application/x-tar',
'xht' => 'application/xhtml+xml',
'xhtml' => 'application/xhtml+xml',
'zip' => 'application/zip',
'mid' => 'audio/midi',
'midi' => 'audio/midi',
'mp2' => 'audio/mpeg',
'mp3' => 'audio/mpeg',
'm4a' => 'audio/mp4',
'mpga' => 'audio/mpeg',
'aif' => 'audio/x-aiff',
'aifc' => 'audio/x-aiff',
'aiff' => 'audio/x-aiff',
'ram' => 'audio/x-pn-realaudio',
'rm' => 'audio/x-pn-realaudio',
'rpm' => 'audio/x-pn-realaudio-plugin',
```

```
'ra' => 'audio/x-realaudio',
'wav' => 'audio/x-wav',
'mka' => 'audio/x-matroska',
'bmp' => 'image/bmp',
'gif' => 'image/gif',
'jpeg' => 'image/jpeg',
'jpe' => 'image/jpeg',
'jpg' => 'image/jpeg',
'png' => 'image/png',
'tiff' => 'image/tiff',
'tif' => 'image/tiff',
'webp' => 'image/webp',
'avif' => 'image/avif',
'heif' => 'image/heif',
'heifs' => 'image/heif-sequence',
'heic' => 'image/heic',
'heics' => 'image/heic-sequence',
'eml' => 'message/rfc822',
'css' => 'text/css',
'html' => 'text/html',
'htm' => 'text/html',
'shtml' => 'text/html',
'log' => 'text/plain',
'text' => 'text/plain',
'txt' => 'text/plain',
'rtx' => 'text/richtext',
'rtf' => 'text/rtf',
'vcf' => 'text/vcard',
'vcard' => 'text/vcard',
'ics' => 'text/calendar',
'xml' => 'text/xml',
'xsl' => 'text/xml',
'csv' => 'text/csv',
'wmv' => 'video/x-ms-wmv',
'mpeg' => 'video/mpeg',
'mpe' => 'video/mpeg',
'mpg' => 'video/mpeg',
'mp4' => 'video/mp4',
'm4v' => 'video/mp4',
'mov' => 'video/quicktime',
'qt' => 'video/quicktime',
'rv' => 'video/vnd.rn-realvideo',
```

```

'avi' => 'video/x-msvideo',
'movie' => 'video/x-sgi-movie',
'webm' => 'video/webm',
'mkv' => 'video/x-matroska',
];
$ext = strtolower($ext);
if (array_key_exists($ext, $mimes)) {
    return $mimes[$ext];
}

return 'application/octet-stream';
}

/** 
 * Map a file name to a MIME type.
 * Defaults to 'application/octet-stream', i.e.. arbitrary binary data.
 *
 * @param string $filename A file name or full path, does not need to exist as
a file
 *
 * @return string
 */
public static function filenameToType($filename)
{
    //In case the path is a URL, strip any query string before getting extension
    $qpos = strpos($filename, '?');
    if (false !== $qpos) {
        $filename = substr($filename, 0, $qpos);
    }
    $ext = static::mb_pathinfo($filename, PATHINFO_EXTENSION);

    return static::_mime_types($ext);
}

/** 
 * Multi-byte-safe pathinfo replacement.
 * Drop-in replacement for pathinfo(), but multibyte- and cross-platform-safe.
 *
 * @see https://www.php.net/manual/en/function.pathinfo.php#107461
 *
 * @param string $path A filename or path, does not need to exist as a
file

```



```

/**
 * Set or reset instance properties.
 * You should avoid this function - it's more verbose, less efficient, more error-prone and
 * harder to debug than setting properties directly.
 * Usage Example:
 * '$mail->set('SMTPSecure', static::ENCRYPTION_STARTTLS);'
 * is the same as:
 * '$mail->SMTPSecure = static::ENCRYPTION_STARTTLS;'.
 *
 * @param string $name The property name to set
 * @param mixed $value The value to set the property to
 *
 * @return bool
 */
public function set($name, $value = "")
{
    if (property_exists($this, $name)) {
        $this->{$name} = $value;

        return true;
    }
    $this->setError($this->lang('variable_set') . $name);

    return false;
}

/**
 * Strip newlines to prevent header injection.
 *
 * @param string $str
 *
 * @return string
 */
public function secureHeader($str)
{
    return trim(str_replace(["\r", "\n"], " ", $str));
}

/**
 * Normalize line breaks in a string.

```

```

    * Converts UNIX LF, Mac CR and Windows CRLF line breaks into a single
line break format.
    * Defaults to CRLF (for message bodies) and preserves consecutive breaks.
    *
    * @param string $text
    * @param string $breaktype What kind of line break to use; defaults to
static::$LE
    *
    * @return string
    */
public static function normalizeBreaks($text, $breaktype = null)
{
    if (null === $breaktype) {
        $breaktype = static::$LE;
    }
    //Normalise to \n
    $text = str_replace([self::CRLF, "\r"], "\n", $text);
    //Now convert LE as needed
    if ("\n" !== $breaktype) {
        $text = str_replace("\n", $breaktype, $text);
    }

    return $text;
}

/**
 * Remove trailing whitespace from a string.
 *
 * @param string $text
 *
 * @return string The text to remove whitespace from
 */
public static function stripTrailingWSP($text)
{
    return rtrim($text, " \r\n\t");
}

/**
 * Strip trailing line breaks from a string.
 *
 * @param string $text
 *

```

```

* @return string The text to remove breaks from
*/
public static function stripTrailingBreaks($text)
{
    return rtrim($text, "\r\n");
}

/** 
 * Return the current line break format string.
 *
 * @return string
 */
public static function getLE()
{
    return static::$LE;
}

/** 
 * Set the line break format string, e.g. "\r\n".
 *
 * @param string $le
 */
protected static function setLE($le)
{
    static::$LE = $le;
}

/** 
 * Set the public and private key files and password for S/MIME signing.
 *
 * @param string $cert_filename
 * @param string $key_filename
 * @param string $key_pass      Password for private key
 * @param string $extracerts_filename Optional path to chain certificate
 */
public function sign($cert_filename, $key_filename, $key_pass,
$extracerts_filename = "")
{
    $this->sign_cert_file = $cert_filename;
    $this->sign_key_file = $key_filename;
    $this->sign_key_pass = $key_pass;
    $this->sign_extracerts_file = $extracerts_filename;
}

```

```

}

/** 
 * Quoted-Printable-encode a DKIM header.
 *
 * @param string $txt
 *
 * @return string
 */
public function DKIM_QP($txt)
{
    $line = "";
    $len = strlen($txt);
    for ($i = 0; $i < $len; ++$i) {
        $ord = ord($txt[$i]);
        if ((0x21 <= $ord) && ($ord <= 0x3A)) || $ord === 0x3C || ((0x3E <= $ord)
&& ($ord <= 0x7E))) {
            $line .= $txt[$i];
        } else {
            $line .= '=' . sprintf('%02X', $ord);
        }
    }

    return $line;
}

/** 
 * Generate a DKIM signature.
 *
 * @param string $signHeader
 *
 * @throws Exception
 *
 * @return string The DKIM signature value
 */
public function DKIM_Sign($signHeader)
{
    if (!defined('PKCS7_TEXT')) {
        if ($this->exceptions) {
            throw new Exception($this->lang('extension_missing') . 'openssl');
        }
    }
}

```

```

        return "";
    }
    $privKeyStr = !empty($this->DKIM_private_string) ?
        $this->DKIM_private_string :
        file_get_contents($this->DKIM_private);
    if ("!==" $this->DKIM_passphrase) {
        $privKey      =      openssl_pkey_get_private($privKeyStr,      $this-
>DKIM_passphrase);
    } else {
        $privKey = openssl_pkey_get_private($privKeyStr);
    }
    if      (openssl_sign($signHeader,      $signature,      $privKey,
'sha256WithRSAEncryption')) {
        if (\PHP_MAJOR_VERSION < 8) {
            openssl_pkey_free($privKey);
        }

        return base64_encode($signature);
    }
    if (\PHP_MAJOR_VERSION < 8) {
        openssl_pkey_free($privKey);
    }

    return "";
}

/**
 * Generate a DKIM canonicalization header.
 * Uses the 'relaxed' algorithm from RFC6376 section 3.4.2.
 * Canonicalized headers should *always* use CRLF, regardless of mailer
setting.
 *
 * @see https://www.rfc-editor.org/rfc/rfc6376#section-3.4.2
 *
 * @param string $signHeader Header
 *
 * @return string
 */
public function DKIM_HeaderC($signHeader)
{
    //Normalize breaks to CRLF (regardless of the mailer)
    $signHeader = static::normalizeBreaks($signHeader, self::CRLF);
}

```

```

//Unfold header lines
//Note PCRE \s is too broad a definition of whitespace; RFC5322 defines it
as '[\t]'
//@see https://www.rfc-editor.org/rfc/rfc5322#section-2.2
//That means this may break if you do something daft like put vertical tabs
in your headers.
$signHeader = preg_replace('/\r\n[\t]+/', ' ', $signHeader);
//Break headers out into an array
$lines = explode(self::CRLF, $signHeader);
foreach ($lines as $key => $line) {
    //If the header is missing a :, skip it as it's invalid
    //This is likely to happen because the explode() above will also split
    //on the trailing LE, leaving an empty line
    if (strpos($line, ':') === false) {
        continue;
    }
    list($heading, $value) = explode(':', $line, 2);
    //Lower-case header name
    $heading = strtolower($heading);
    //Collapse white space within the value, also convert WSP to space
    $value = preg_replace('/[\t]+/', ' ', $value);
    //RFC6376 is slightly unclear here - it says to delete space at the *end*
of each value
        //But then says to delete space before and after the colon.
        //Net result is the same as trimming both ends of the value.
        //By elimination, the same applies to the field name
        $lines[$key] = trim($heading, " \t") . ':' . trim($value, " \t");
    }
}

return implode(self::CRLF, $lines);
}

/**
 * Generate a DKIM canonicalization body.
 * Uses the 'simple' algorithm from RFC6376 section 3.4.3.
 * Canonicalized bodies should *always* use CRLF, regardless of mailer
setting.
 *
 * @see https://www.rfc-editor.org/rfc/rfc6376#section-3.4.3
 *
 * @param string $body Message Body
 *

```

```

* @return string
*/
public function DKIM_BodyC($body)
{
    if (empty($body)) {
        return self::CRLF;
    }
    //Normalize line endings to CRLF
    $body = static::normalizeBreaks($body, self::CRLF);

    //Reduce multiple trailing line breaks to a single one
    return static::stripTrailingBreaks($body) . self::CRLF;
}

/**
 * Create the DKIM header and body in a new message header.
 *
 * @param string $headers_line Header lines
 * @param string $subject      Subject
 * @param string $body         Body
 *
 * @throws Exception
 *
 * @return string
 */
public function DKIM_Add($headers_line, $subject, $body)
{
    $DKIMsignatureType = 'rsa-sha256'; //Signature & hash algorithms
    $DKIMcanonicalization = 'relaxed/simple'; //Canonicalization methods of
header & body
    $DKIMquery = 'dns/txt'; //Query method
    $DKIMtime = time();
    //Always sign these headers without being asked
    //Recommended list from https://www.rfc-editor.org/rfc/rfc6376#section-5.4.1
    $autoSignHeaders = [
        'from',
        'to',
        'cc',
        'date',
        'subject',
        'reply-to',
    ];
}

```

```

'message-id',
'content-type',
'mime-version',
'x-mailer',
];
if (stripos($headers_line, 'Subject') === false) {
    $headers_line .= 'Subject: ' . $subject . static::$LE;
}
$headerLines = explode(static::$LE, $headers_line);
$currentHeaderLabel = "";
$currentHeaderValue = "";
$parsedHeaders = [];
$headerLineIndex = 0;
$headerLineCount = count($headerLines);
foreach ($headerLines as $headerLine) {
    $matches = [];
    if (preg_match('/^([^\t]*?)(?:[ \t]*)(.*$)/', $headerLine, $matches)) {
        if ($currentHeaderLabel !== "") {
            //We were previously in another header; This is the start of a new
            header, so save the previous one
            $parsedHeaders[] = ['label' => $currentHeaderLabel, 'value' =>
$currentHeaderValue];
        }
        $currentHeaderLabel = $matches[1];
        $currentHeaderValue = $matches[2];
    } elseif (preg_match('/^[\t]+(.*)$/', $headerLine, $matches)) {
        //This is a folded continuation of the current header, so unfold it
        $currentHeaderValue .= ' ' . $matches[1];
    }
    ++$headerLineIndex;
    if ($headerLineIndex >= $headerLineCount) {
        //This was the last line, so finish off this header
        $parsedHeaders[] = ['label' => $currentHeaderLabel, 'value' =>
$currentHeaderValue];
    }
}
$copiedHeaders = [];
$headersToSignKeys = [];
$headersToSign = [];
foreach ($parsedHeaders as $header) {
    //Is this header one that must be included in the DKIM signature?
    if (in_array(strtolower($header['label']), $autoSignHeaders, true)) {

```

```

$headersToSignKeys[] = $header['label'];
$headersToSign[] = $header['label'] . ':' . $header['value'];
if ($this->DKIM_copyHeaderFields) {
    $copiedHeaders[] = $header['label'] . ':' . //Note no space after this,
as per RFC
        str_replace('|', '=7C', $this->DKIM_QP($header['value']));
}
continue;
}
//Is this an extra custom header we've been asked to sign?
if (in_array($header['label'], $this->DKIM_extraHeaders, true)) {
    //Find its value in custom headers
foreach ($this->CustomHeader as $customHeader) {
    if ($customHeader[0] === $header['label']) {
        $headersToSignKeys[] = $header['label'];
        $headersToSign[] = $header['label'] . ':' . $header['value'];
        if ($this->DKIM_copyHeaderFields) {
            $copiedHeaders[] = $header['label'] . ':' . //Note no space after
this, as per RFC
                str_replace('|', '=7C', $this->DKIM_QP($header['value']));
            }
        //Skip straight to the next header
        continue 2;
    }
}
}
$copiedHeaderFields = "";
if ($this->DKIM_copyHeaderFields && count($copiedHeaders) > 0) {
    //Assemble a DKIM 'z' tag
    $copiedHeaderFields = ' z=';
    $first = true;
    foreach ($copiedHeaders as $copiedHeader) {
        if (!$first) {
            $copiedHeaderFields .= static::$LE . '|';
        }
        //Fold long values
        if (strlen($copiedHeader) > self::STD_LINE_LENGTH - 3) {
            $copiedHeaderFields .= substr(
                chunk_split($copiedHeader,  self::STD_LINE_LENGTH - 3,
static::$LE . self::FWS),
                0,

```

```

        -strlen(static::$LE . self::FWS)
    );
} else {
    $copiedHeaderFields .= $copiedHeader;
}
$first = false;
}
$copiedHeaderFields .= ';' . static::$LE;
}

$headerKeys = ' h=' . implode(':', $headersToSignKeys) . ';' . static::$LE;
$headerValues = implode(static::$LE, $headersToSign);
$body = $this->DKIM_BodyC($body);
//Base64 of packed binary SHA-256 hash of body
$DKIMb64 = base64_encode(pack('H*', hash('sha256', $body)));
$ident = '';
if ("!==" . $this->DKIM_identity) {
    $ident = ' i=' . $this->DKIM_identity . ';' . static::$LE;
}
//The DKIM-Signature header is included in the signature *except for* the
value of the `b` tag
//which is appended after calculating the signature
//https://www.rfc-editor.org/rfc/rfc6376#section-3.5
$dkimSignatureHeader = 'DKIM-Signature: v=1,' .
    ' d=' . $this->DKIM_domain . ';' .
    ' s=' . $this->DKIM_selector . ';' . static::$LE .
    ' a=' . $DKIMsignatureType . ';' .
    ' q=' . $DKIMquery . ';' .
    ' t=' . $DKIMtime . ';' .
    ' c=' . $DKIMcanonicalization . ';' . static::$LE .
$headerKeys .
$ident .
$copiedHeaderFields .
' bh=' . $DKIMb64 . ';' . static::$LE .
' b=';

//Canonicalize the set of headers
$canonicalizedHeaders = $this->DKIM_HeaderC(
    $headerValues . static::$LE . $dkimSignatureHeader
);
$signature = $this->DKIM_Sign($canonicalizedHeaders);
$signature = trim(chunk_split($signature, self::STD_LINE_LENGTH - 3,
static::$LE . self::FWS));

```

```

        return static::normalizeBreaks($dkimSignatureHeader . $signature);
    }

    /**
     * Detect if a string contains a line longer than the maximum line length
     * allowed by RFC 2822 section 2.1.1.
     *
     * @param string $str
     *
     * @return bool
     */
    public static function hasLineLongerThanMax($str)
    {
        return (bool) preg_match('/^(.{' . (self::MAX_LINE_LENGTH + strlen(static::$LE)) . ',})/m', $str);
    }

    /**
     * If a string contains any "special" characters, double-quote the name,
     * and escape any double quotes with a backslash.
     *
     * @param string $str
     *
     * @return string
     *
     * @see RFC822 3.4.1
     */
    public static function quotedString($str)
    {
        if (preg_match('/[ ()<>@,:"\\"[\"]?=]/', $str)) {
            //If the string contains any of these chars, it must be double-quoted
            //and any double quotes must be escaped with a backslash
            return "" . str_replace("", '\\', $str) . "";
        }

        //Return the string untouched, it doesn't need quoting
        return $str;
    }

    /**
     * Allows for public read access to 'to' property.
     * Before the send() call, queued addresses (i.e. with IDN) are not yet included.

```

```

/*
 * @return array
 */
public function getToAddresses()
{
    return $this->to;
}

/** 
 * Allows for public read access to 'cc' property.
 * Before the send() call, queued addresses (i.e. with IDN) are not yet included.
 *
 * @return array
 */
public function getCcAddresses()
{
    return $this->cc;
}

/** 
 * Allows for public read access to 'bcc' property.
 * Before the send() call, queued addresses (i.e. with IDN) are not yet included.
 *
 * @return array
 */
public function getBccAddresses()
{
    return $this->bcc;
}

/** 
 * Allows for public read access to 'ReplyTo' property.
 * Before the send() call, queued addresses (i.e. with IDN) are not yet included.
 *
 * @return array
 */
public function getReplyToAddresses()
{
    return $this->ReplyTo;
}

*/

```

```

* Allows for public read access to 'all_recipients' property.
* Before the send() call, queued addresses (i.e. with IDN) are not yet included.
*
* @return array
*/
public function getAllRecipientAddresses()
{
    return $this->all_recipients;
}

/**
* Perform a callback.
*
* @param bool $isSent
* @param array $to
* @param array $cc
* @param array $bcc
* @param string $subject
* @param string $body
* @param string $from
* @param array $extra
*/
protected function doCallback($isSent, $to, $cc, $bcc, $subject, $body, $from,
$extra)
{
    if (!empty($this->action_function) && is_callable($this->action_function)) {
        call_user_func($this->action_function, $isSent, $to, $cc, $bcc, $subject,
$body, $from, $extra);
    }
}

/**
* Get the OAuthTokenProvider instance.
*
* @return OAuthTokenProvider
*/
public function getOAuth()
{
    return $this->oauth;
}

/**

```

```

    * Set an OAuthTokenProvider instance.
    */
public function setOAuth(OAuthTokenProvider $oauth)
{
    $this->oauth = $oauth;
}
}

|   |— SMTP.php

<?php

/**
 * PHPMailer RFC821 SMTP email transport class.
 * PHP Version 5.5.
 *
 * @see      https://github.com/PHPMailer/PHPMailer/ The PHPMailer
GitHub project
 *
 * @author  Marcus Bointon (Synchro/coolbru)
<phpmailer@synchromedia.co.uk>
 * @author  Jim Jagielski (jimjag) <jimjag@gmail.com>
 * @author  Andy Prevost (codeworxtech)
<codeworxtech@users.sourceforge.net>
 * @author  Brent R. Matzelle (original founder)
 * @copyright 2012 - 2020 Marcus Bointon
 * @copyright 2010 - 2012 Jim Jagielski
 * @copyright 2004 - 2009 Andy Prevost
 * @license  https://www.gnu.org/licenses/lgpl-2.1.html GNU
Lesser General Public License
 * @note This program is distributed in the hope that it will be
useful - WITHOUT
 * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
 * FITNESS FOR A PARTICULAR PURPOSE.
 */

namespace PHPMailer\PHPMailer;

/**
 * PHPMailer RFC821 SMTP email transport class.
 * Implements RFC 821 SMTP commands and provides some utility methods for
sending mail to an SMTP server.
 *
 * @author Chris Ryan
 * @author Marcus Bointon <phpmailer@synchromedia.co.uk>
 */
class SMTP
{
    /**

```

```

 * The PHPMailer SMTP version number.
 *
 * @var string
 */
const VERSION = '6.10.0';

/**
 * SMTP line break constant.
 *
 * @var string
 */
const LE = "\r\n";

/**
 * The SMTP port to use if one is not specified.
 *
 * @var int
 */
const DEFAULT_PORT = 25;

/**
 * The SMTPs port to use if one is not specified.
 *
 * @var int
 */
const DEFAULT_SECURE_PORT = 465;

/**
 * The maximum line length allowed by RFC 5321 section 4.5.3.1.6,
 * *excluding* a trailing CRLF break.
 *
 * @see https://www.rfc-editor.org/rfc/rfc5321#section-4.5.3.1.6
 *
 * @var int
 */
const MAX_LINE_LENGTH = 998;

/**
 * The maximum line length allowed for replies in RFC 5321 section
4.5.3.1.5,
 * *including* a trailing CRLF line break.
 *
 * @see https://www.rfc-editor.org/rfc/rfc5321#section-4.5.3.1.5
 *
 * @var int
 */
const MAX_REPLY_LENGTH = 512;

```

```

/**
 * Debug level for no output.
 *
 * @var int
 */
const DEBUG_OFF = 0;

/**
 * Debug level to show client -> server messages.
 *
 * @var int
 */
const DEBUG_CLIENT = 1;

/**
 * Debug level to show client -> server and server -> client
messages.
 *
 * @var int
 */
const DEBUG_SERVER = 2;

/**
 * Debug level to show connection status, client -> server and server
-> client messages.
 *
 * @var int
 */
const DEBUG_CONNECTION = 3;

/**
 * Debug level to show all messages.
 *
 * @var int
 */
const DEBUG_LOWLEVEL = 4;

/**
 * Debug output level.
 * Options:
 * * self::DEBUG_OFF (`0`) No debug output, default
 * * self::DEBUG_CLIENT (`1`) Client commands
 * * self::DEBUG_SERVER (`2`) Client commands and server responses
 * * self::DEBUG_CONNECTION (`3`) As DEBUG_SERVER plus connection
status
 * * self::DEBUG_LOWLEVEL (`4`) Low-level data output, all messages.
 *
 * @var int

```

```

*/
public $do_debug = self::DEBUG_OFF;

/**
 * How to handle debug output.
 * Options:
 *  * `echo` Output plain-text as-is, appropriate for CLI
 *  * `html` Output escaped, line breaks converted to `<br>`,  

appropriate for browser output
 *  * `error_log` Output to error log as configured in php.ini
 * Alternatively, you can provide a callable expecting two params: a  

message string and the debug level:
 *
 * ``php
 * $smtp->Debugoutput = function($str, $level) {echo "debug level  

$level; message: $str";};
 * ``

 *
 * Alternatively, you can pass in an instance of a PSR-3 compatible  

logger, though only `debug`
 *  * level output is used:
 *
 * ``php
 * $mail->Debugoutput = new myPsr3Logger;
 * ``

 *
 * @var string|callable|\Psr\Log\LoggerInterface
 */
public $Debugoutput = 'echo';

/**
 * Whether to use VERP.
 *
 * @see https://en.wikipedia.org/wiki/Variable_envelope_return_path
 * @see https://www.postfix.org/VERP_README.html Info on VERP
 *
 * @var bool
 */
public $do_verp = false;

/**
 * Whether to use SMTPUTF8.
 *
 * @see https://www.rfc-editor.org/rfc/rfc6531
 *
 * @var bool
 */
public $do_smtputf8 = false;

```

```

/**
 * The timeout value for connection, in seconds.
 * Default of 5 minutes (300sec) is from RFC2821 section 4.5.3.2.
 * This needs to be quite high to function correctly with hosts using
greetdelay as an anti-spam measure.
 *
 * @see https://www.rfc-editor.org/rfc/rfc2821#section-4.5.3.2
 *
 * @var int
 */
public $Timeout = 300;

/**
 * How long to wait for commands to complete, in seconds.
 * Default of 5 minutes (300sec) is from RFC2821 section 4.5.3.2.
 *
 * @var int
 */
public $Timelimit = 300;

/**
 * Patterns to extract an SMTP transaction id from reply to a DATA
command.
 * The first capture group in each regex will be used as the ID.
 * MS ESMTP returns the message ID, which may not be correct for
internal tracking.
 *
 * @var string[]
 */
protected $smtp_transaction_id_patterns = [
    'exim' => '/[\d]{3} OK id=(.*)/',
    'sendmail' => '/[\d]{3} 2\.0\.0 (.*) Message/',
    'postfix' => '/[\d]{3} 2\.0\.0 Ok: queued as (.*)/',
    'Microsoft_ESMTP' => '/[0-9]{3} 2\.[\d]\.0 (.*)@(?:.*)
mail for delivery/',
    'Amazon_SES' => '/[\d]{3} Ok (.*)/',
    'SendGrid' => '/[\d]{3} Ok: queued as (.*)/',
    'CampaignMonitor' => '/[\d]{3} 2\.0\.0 OK:([a-zA-Z\d]{48})/',
    'Haraka' => '/[\d]{3} Message Queued \((.*)\)/',
    'ZoneMTA' => '/[\d]{3} Message queued as (.*)/',
    'Mailjet' => '/[\d]{3} OK queued as (.*)/',
];
}

/**
 * Allowed SMTP XCLIENT attributes.
 * Must be allowed by the SMTP server. EHLO response is not checked.
 *

```

```

* @see https://www.postfix.org/XCLIENT\_README.html
*
* @var array
*/
public static $xclient_allowed_attributes = [
    'NAME', 'ADDR', 'PORT', 'PROTO', 'HELO', 'LOGIN', 'DESTADDR',
'DESTPORT'
];

/**
 * The last transaction ID issued in response to a DATA command,
* if one was detected.
*
* @var string|bool|null
*/
protected $last_smtp_transaction_id;

/**
 * The socket for the server connection.
*
* @var ?resource
*/
protected $smtp_conn;

/**
 * Error information, if any, for the last SMTP command.
*
* @var array
*/
protected $error = [
    'error' => '',
    'detail' => '',
    'smtp_code' => '',
    'smtp_code_ex' => '',
];
;

/**
 * The reply the server sent to us for HELO.
* If null, no HELO string has yet been received.
*
* @var string|null
*/
protected $helo_rply;

/**
 * The set of SMTP extensions sent in reply to EHLO command.
* Indexes of the array are extension names.

```

```

        * Value at index 'HELO' or 'EHLO' (according to command that was
sent)
        * represents the server name. In case of HELO it is the only element
of the array.
        * Other values can be boolean TRUE or an array containing extension
options.
        * If null, no HELO/EHLO string has yet been received.
        *
        * @var array|null
        */
protected $server_caps;

/**
 * The most recent reply received from the server.
 *
 * @var string
 */
protected $last_reply = '';

/**
 * Output debugging info via a user-selected method.
 *
 * @param string $str    Debug string to output
 * @param int    $level   The debug level of this message; see DEBUG_*
constants
 *
 * @see SMTP::$Debugoutput
 * @see SMTP:$do_debug
 */
protected function edebug($str, $level = 0)
{
    if ($level > $this->do_debug) {
        return;
    }
    //Is this a PSR-3 logger?
    if ($this->Debugoutput instanceof \Psr\Log\LoggerInterface) {
        //Remove trailing line breaks potentially added by calls to
SMTP::client_send()
        $this->Debugoutput->debug(rtrim($str, "\r\n"));

        return;
    }
    //Avoid clash with built-in function names
    if (is_callable($this->Debugoutput) && !in_array($this-
>Debugoutput, ['error_log', 'html', 'echo'])) {
        call_user_func($this->Debugoutput, $str, $level);

        return;
    }
}

```

```
}

switch ($this->Debugoutput) {
    case 'error_log':
        //Don't output, just log
        /** @noinspection ForgottenDebugOutputInspection */
        error_log($str);
        break;
    case 'html':
        //Cleans up output a bit for a better looking, HTML-safe
output
        echo gmdate('Y-m-d H:i:s'), ' ', htmlentities(
            preg_replace('/[\r\n]+/', ' ', $str),
            ENT_QUOTES,
            'UTF-8'
        ), "<br>\n";
        break;
    case 'echo':
    default:
        //Normalize line breaks
        $str = preg_replace('/\r\n|\r/m', "\n", $str);
        echo gmdate('Y-m-d H:i:s'),
        "\t",
        //Trim trailing space
        trim(
            //Indent for readability, except for trailing break
            str_replace(
                "\n",
                "\n\t",
                trim($str)
            )
        ),
        "\n";
}
}

/**
 * Connect to an SMTP server.
 *
 * @param string $host      SMTP server IP or host name
 * @param int    $port      The port number to connect to
 * @param int    $timeout   How long to wait for the connection to open
 * @param array  $options   An array of options for
stream_context_create()
*
* @return bool
*/
public function connect($host, $port = null, $timeout = 30, $options
= [])
{
    $context = stream_context_create([
        'ssl' => [
            'verify_peer' => false,
            'verify_peer_name' => false
        ]
    ]);

    if ($port === null) {
        $port = 587;
    }

    if ($host === null) {
        $host = 'localhost';
    }

    if ($timeout === null) {
        $timeout = 30;
    }

    if ($options === null) {
        $options = [];
    }

    $options['socket_timeout'] = $timeout;

    if (!empty($options)) {
        $context = stream_context_create($options);
    }

    return stream_socket_client("tls://{$host}:{$port}", $err, $errstr, 30, STREAM_CLIENT_CONNECT | STREAM_CLIENT_PERSISTENT, $context);
}
```

```

{
    //Clear errors to avoid confusion
    $this->setError('');
    //Make sure we are __not__ connected
    if ($this->connected()) {
        //Already connected, generate error
        $this->setError('Already connected to a server');

        return false;
    }
    if (empty($port)) {
        $port = self::DEFAULT_PORT;
    }
    //Connect to the SMTP server
    $this->edebug(
        "Connection: opening to $host:$port, timeout=$timeout,
options=" .
        (count($options) > 0 ? var_export($options, true) :
'array()'),
        self::DEBUG_CONNECTION
    );

    $this->smtp_conn = $this->getSMTPConnection($host, $port,
$timeout, $options);

    if ($this->smtp_conn === false) {
        //Error info already set inside `getSMTPConnection()`
        return false;
    }

    $this->edebug('Connection: opened', self::DEBUG_CONNECTION);

    //Get any announcement
    $this->last_reply = $this->get_lines();
    $this->edebug('SERVER -> CLIENT: ' . $this->last_reply,
self::DEBUG_SERVER);
    $responseCode = (int)substr($this->last_reply, 0, 3);
    if ($responseCode === 220) {
        return true;
    }
    //Anything other than a 220 response means something went wrong
    //RFC 5321 says the server will wait for us to send a QUIT in
response to a 554 error
    //https://www.rfc-editor.org/rfc/rfc5321#section-3.1
    if ($responseCode === 554) {
        $this->quit();
    }
}

```

```

        //This will handle 421 responses which may not wait for a QUIT
        (e.g. if the server is being shut down)
            $this->edebug('Connection: closing due to error',
self::DEBUG_CONNECTION);
            $this->close();
            return false;
        }

    /**
     * Create connection to the SMTP server.
     *
     * @param string $host      SMTP server IP or host name
     * @param int    $port      The port number to connect to
     * @param int    $timeout   How long to wait for the connection to open
     * @param array  $options   An array of options for
stream_context_create()
*
* @return false|resource
*/
protected function getSMTPConnection($host, $port = null, $timeout =
30, $options = [])
{
    static $streamok;
    //This is enabled by default since 5.0.0 but some providers
disable it
    //Check this once and cache the result
    if (null === $streamok) {
        $streamok = function_exists('stream_socket_client');
    }

    $errno = 0;
    $errstr = '';
    if ($streamok) {
        $socket_context = stream_context_create($options);
        set_error_handler(function () {
            call_user_func_array([$this, 'errorHandler'],
func_get_args()));
        });
        $connection = stream_socket_client(
            $host . ':' . $port,
            $errno,
            $errstr,
            $timeout,
            STREAM_CLIENT_CONNECT,
            $socket_context
        );
    } else {

```

```

        //Fall back to fsockopen which should work in more places,
but is missing some features
        $this->edebug(
            'Connection: stream_socket_client not available, falling
back to fsockopen',
            self::DEBUG_CONNECTION
        );
        set_error_handler(function () {
            call_user_func_array([$this, 'errorHandler'],
func_get_args()));
        });
        $connection = fsockopen(
            $host,
            $port,
            $errno,
            $errstr,
            $timeout
        );
    }
    restore_error_handler();

    //Verify we connected properly
    if (!is_resource($connection)) {
        $this->setError(
            'Failed to connect to server',
            '',
            (string) $errno,
            $errstr
        );
        $this->edebug(
            'SMTP ERROR: ' . $this->error['error']
            . ": $errstr ($errno)",
            self::DEBUG_CLIENT
        );
        return false;
    }

    //SMTP server can take longer to respond, give longer timeout for
first read
    //Windows does not have support for this timeout function
    if (strpos(PHP_OS, 'WIN') !== 0) {
        $max = (int)ini_get('max_execution_time');
        //Don't bother if unlimited, or if set_time_limit is disabled
        if (0 !== $max && $timeout > $max &&
strpos(ini_get('disable_functions'), 'set_time_limit') === false) {
            @set_time_limit($timeout);
        }
    }
}

```

```

        stream_set_timeout($connection, $timeout, 0);
    }

    return $connection;
}

/***
 * Initiate a TLS (encrypted) session.
 *
 * @return bool
 */
public function startTLS()
{
    if (!$this->sendCommand('STARTTLS', 'STARTTLS', 220)) {
        return false;
    }

    //Allow the best TLS version(s) we can
$crypto_method = STREAM_CRYPTO_METHOD_TLS_CLIENT;

    //PHP 5.6.7 dropped inclusion of TLS 1.1 and 1.2 in
STREAM_CRYPTO_METHOD_TLS_CLIENT
    //so add them back in manually if we can
    if (defined('STREAM_CRYPTO_METHOD_TLSv1_2_CLIENT')) {
        $crypto_method |= STREAM_CRYPTO_METHOD_TLSv1_2_CLIENT;
        $crypto_method |= STREAM_CRYPTO_METHOD_TLSv1_1_CLIENT;
    }

    //Begin encrypted connection
    set_error_handler(function () {
        call_user_func_array([$this, 'errorHandler'],
func_get_args());
    });
    $crypto_ok = stream_socket_enable_crypto(
        $this->smtp_conn,
        true,
        $crypto_method
    );
    restore_error_handler();

    return (bool) $crypto_ok;
}

/***
 * Perform SMTP authentication.
 * Must be run after hello().
 *
 * @see     hello()
*/

```

```

*
 * @param string $username The user name
 * @param string $password The password
 * @param string $authtype The auth type (CRAM-MD5, PLAIN, LOGIN,
XOAUTH2)
 * @param OAuthTokenProvider $OAuth An optional OAuthTokenProvider
instance for XOAUTH2 authentication
*
* @return bool True if successfully authenticated
*/
public function authenticate(
    $username,
    $password,
    $authtype = null,
    $OAuth = null
) {
    if (!$this->server_caps) {
        $this->setError('Authentication is not allowed before
HELO/EHLO');

        return false;
    }

    if (array_key_exists('EHLO', $this->server_caps)) {
        //SMTP extensions are available; try to find a proper
authentication method
        if (!array_key_exists('AUTH', $this->server_caps)) {
            $this->setError('Authentication is not allowed at this
stage');

            //at this stage' means that auth may be allowed after
the stage changes
            //e.g. after STARTTLS

            return false;
        }

        $this->edebug('Auth method requested: ' . ($authtype ?:
'UNSPECIFIED'), self::DEBUG_LOWLEVEL);
        $this->edebug(
            'Auth methods available on the server: ' . implode(',',
$this->server_caps['AUTH']),
            self::DEBUG_LOWLEVEL
        );
    }

    //If we have requested a specific auth type, check the server
supports it before trying others
    if (null !== $authtype && !in_array($authtype, $this-
>server_caps['AUTH'], true)) {

```

```

        $this->edebug('Requested auth method not available: ' . $authtype, self::DEBUG_LOWLEVEL);
                $authtype = null;
}

if (empty($authtype)) {
    //If no auth mechanism is specified, attempt to use
these, in this order
    //Try CRAM-MD5 first as it's more secure than the others
    foreach (['CRAM-MD5', 'LOGIN', 'PLAIN', 'XOAUTH2'] as
$method) {
        if (in_array($method, $this->server_caps['AUTH'],
true)) {
            $authtype = $method;
            break;
        }
    }
    if (empty($authtype)) {
        $this->setError('No supported authentication methods
found');

        return false;
    }
    $this->edebug('Auth method selected: ' . $authtype,
self::DEBUG_LOWLEVEL);
}

if (!in_array($authtype, $this->server_caps['AUTH'], true)) {
    $this->setError("The requested authentication method
\"$authtype\" is not supported by the server");

    return false;
}
} elseif (empty($authtype)) {
    $authtype = 'LOGIN';
}
switch ($authtype) {
    case 'PLAIN':
        //Start authentication
        if (!$this->sendCommand('AUTH', 'AUTH PLAIN', 334)) {
            return false;
        }
        //Send encoded username and password
        if (
            //Format from https://www.rfc-
editor.org/rfc/rfc4616#section-2
            //We skip the first field (it's forgery), so the
            string starts with a null byte

```

```

        !$this->sendCommand(
            'User & Password',
            base64_encode("\0" . $username . "\0" .
$password),
            235
        )
    ) {
        return false;
    }
    break;
case 'LOGIN':
    //Start authentication
    if (!$this->sendCommand('AUTH', 'AUTH LOGIN', 334)) {
        return false;
    }
    if (!$this->sendCommand('Username',
base64_encode($username), 334)) {
        return false;
    }
    if (!$this->sendCommand('Password',
base64_encode($password), 235)) {
        return false;
    }
    break;
case 'CRAM-MD5':
    //Start authentication
    if (!$this->sendCommand('AUTH CRAM-MD5', 'AUTH CRAM-MD5',
334)) {
        return false;
    }
    //Get the challenge
    $challenge = base64_decode(substr($this->last_reply, 4));

    //Build the response
    $response = $username . ' ' . $this->hmac($challenge,
$password);

    //Send encoded credentials
    return $this->sendCommand('Username',
base64_encode($response), 235);
case 'XOAUTH2':
    //The OAuth instance must be set up prior to requesting
auth.
    if (null === $OAuth) {
        return false;
    }
    $oauth = $OAuth->getOauth64();

```

```

//Start authentication
if (!$this->sendCommand('AUTH', 'AUTH XOAUTH2 ' . $oauth,
235)) {
    return false;
}
break;
default:
    $this->setError("Authentication method \\"$authtype\\" is
not supported");

    return false;
}

return true;
}

/**
 * Calculate an MD5 HMAC hash.
 * Works like hash_hmac('md5', $data, $key)
 * in case that function is not available.
 *
 * @param string $data The data to hash
 * @param string $key The key to hash with
 *
 * @return string
 */
protected function hmac($data, $key)
{
    if (function_exists('hash_hmac')) {
        return hash_hmac('md5', $data, $key);
    }

    //The following borrowed from
    //https://www.php.net/manual/en/function.mhash.php#27225

    //RFC 2104 HMAC implementation for php.
    //Creates an md5 HMAC.
    //Eliminates the need to install mhash to compute a HMAC
    //by Lance Rushing

    $bytelen = 64; //byte length for md5
    if (strlen($key) > $bytelen) {
        $key = pack('H*', md5($key));
    }
    $key = str_pad($key, $bytelen, chr(0x00));
    $ipad = str_pad(' ', $bytelen, chr(0x36));
    $opad = str_pad(' ', $bytelen, chr(0x5c));
    $k_ipad = $key ^ $ipad;
}

```

```

    $k_opad = $key ^ $opad;

    return md5($k_opad . pack('H*', md5($k_ipad . $data)));
}

/***
 * Check connection state.
 *
 * @return bool True if connected
 */
public function connected()
{
    if (is_resource($this->smtp_conn)) {
        $sock_status = stream_get_meta_data($this->smtp_conn);
        if ($sock_status['eof']) {
            //The socket is valid but we are not connected
            $this->edebug(
                'SMTP NOTICE: EOF caught while checking if
connected',
                self::DEBUG_CLIENT
            );
            $this->close();

            return false;
        }

        return true; //everything looks good
    }

    return false;
}

/***
 * Close the socket and clean up the state of the class.
 * Don't use this function without first trying to use QUIT.
 *
 * @see quit()
 */
public function close()
{
    $this->server_caps = null;
    $this->helo_rply = null;
    if (is_resource($this->smtp_conn)) {
        //Close the connection and cleanup
        fclose($this->smtp_conn);
        $this->smtp_conn = null; //Makes for cleaner serialization
        $this->edebug('Connection: closed', self::DEBUG_CONNECTION);
    }
}

```

```

}

/** 
 * Send an SMTP DATA command.
 * Issues a data command and sends the msg_data to the server,
 * finalizing the mail transaction. $msg_data is the message
 * that is to be sent with the headers. Each header needs to be
 * on a single line followed by a <CRLF> with the message headers
 * and the message body being separated by an additional <CRLF>.
 * Implements RFC 821: DATA <CRLF>.
 *
 * @param string $msg_data Message data to send
 *
 * @return bool
 */
public function data($msg_data)
{
    //This will use the standard timelimit
    if (!$this->sendCommand('DATA', 'DATA', 354)) {
        return false;
    }

    /* The server is ready to accept data!
     * According to rfc821 we should not send more than 1000
     characters on a single line (including the LE)
     * so we will break the data up into lines by \r and/or \n then
     if needed we will break each of those into
     * smaller lines to fit within the limit.
     * We will also look for lines that start with a '.' and prepend
     an additional '..'.
     * NOTE: this does not count towards line-length limit.
    */

    //Normalize line breaks before exploding
    $lines = explode("\n", str_replace(["\r\n", "\r"], "\n",
$msg_data));

    /* To distinguish between a complete RFC822 message and a plain
message body, we check if the first field
     * of the first line (':' separated) does not contain a space
then it _should_ be a header, and we will
     * process all lines before a blank line as headers.
    */

    $field = substr($lines[0], 0, strpos($lines[0], ':'));
    $in_headers = false;
    if (!empty($field) && strpos($field, ' ') === false) {
        $in_headers = true;
    }
}

```

```

    }

    foreach ($lines as $line) {
        $lines_out = [];
        if ($in_headers && $line === '') {
            $in_headers = false;
        }
        //Break this line up into several smaller lines if it's too
long
        //Micro-optimisation: isset($str[$len]) is faster than
(strlen($str) > $len),
        while (isset($line[self::MAX_LINE_LENGTH])) {
            //Working backwards, try to find a space within the last
MAX_LINE_LENGTH chars of the line to break on
            //so as to avoid breaking in the middle of a word
            $pos = strrpos(substr($line, 0, self::MAX_LINE_LENGTH), ' ');
        };
        //Deliberately matches both false and 0
        if (!$pos) {
            //No nice break found, add a hard break
            $pos = self::MAX_LINE_LENGTH - 1;
            $lines_out[] = substr($line, 0, $pos);
            $line = substr($line, $pos);
        } else {
            //Break at the found point
            $lines_out[] = substr($line, 0, $pos);
            //Move along by the amount we dealt with
            $line = substr($line, $pos + 1);
        }
        //If processing headers add a LWSP-char to the front of
new line RFC822 section 3.1.1
        if ($in_headers) {
            $line = "\t" . $line;
        }
    }
    $lines_out[] = $line;

    //Send the lines to the server
    foreach ($lines_out as $line_out) {
        //Dot-stuffing as per RFC5321 section 4.5.2
        //https://www.rfc-editor.org/rfc/rfc5321#section-4.5.2
        if (!empty($line_out) && $line_out[0] === '.') {
            $line_out = '.' . $line_out;
        }
        $this->client_send($line_out . static::LE, 'DATA');
    }
}

```

```

//Message data has been sent, complete the command
//Increase timelimit for end of DATA command
$savetimelimit = $this->Timelimit;
$this->Timelimit *= 2;
$result = $this->sendCommand('DATA END', '.', 250);
$this->recordLastTransactionID();
//Restore timelimit
$this->Timelimit = $savetimelimit;

return $result;
}

/**
 * Send an SMTP HELO or EHLO command.
 * Used to identify the sending server to the receiving server.
 * This makes sure that client and server are in a known state.
 * Implements RFC 821: HELO <SP> <domain> <CRLF>
 * and RFC 2821 EHLO.
 *
 * @param string $host The host name or IP to connect to
 *
 * @return bool
 */
public function hello($host = '')
{
    //Try extended hello first (RFC 2821)
    if ($this->sendHello('EHLO', $host)) {
        return true;
    }

    //Some servers shut down the SMTP service here (RFC 5321)
    if (substr($this->helo_rply, 0, 3) == '421') {
        return false;
    }

    return $this->sendHello('HELO', $host);
}

/**
 * Send an SMTP HELO or EHLO command.
 * Low-level implementation used by hello().
 *
 * @param string $hello The HELO string
 * @param string $host The hostname to say we are
 *
 * @return bool
 *
 * @see hello()

```

```

*/
protected function sendHello($hello, $host)
{
    $noerror = $this->sendCommand($hello, $hello . ' ' . $host, 250);
    $this->helo_rply = $this->last_reply;
    if ($noerror) {
        $this->parseHelloFields($hello);
    } else {
        $this->server_caps = null;
    }

    return $noerror;
}

/**
 * Parse a reply to HELO/EHLO command to discover server extensions.
 * In case of HELO, the only parameter that can be discovered is a
server name.
 *
 * @param string $type `HELO` or `EHLO`
 */
protected function parseHelloFields($type)
{
    $this->server_caps = [];
    $lines = explode("\n", $this->helo_rply);

    foreach ($lines as $n => $s) {
        //First 4 chars contain response code followed by - or space
        $s = trim(substr($s, 4));
        if (empty($s)) {
            continue;
        }
        $fields = explode(' ', $s);
        if (!empty($fields)) {
            if (!$n) {
                $name = $type;
                $fields = $fields[0];
            } else {
                $name = array_shift($fields);
                switch ($name) {
                    case 'SIZE':
                        $fields = ($fields ? $fields[0] : 0);
                        break;
                    case 'AUTH':
                        if (!is_array($fields)) {
                            $fields = [];
                        }
                        break;
                }
            }
        }
    }
}

```

```

        default:
            $fields = true;
        }
    }
    $this->server_caps[$name] = $fields;
}
}

/***
 * Send an SMTP MAIL command.
 * Starts a mail transaction from the email address specified in
 * $from. Returns true if successful or false otherwise. If True
 * the mail transaction is started and then one or more recipient
 * commands may be called followed by a data command.
 * Implements RFC 821: MAIL <SP> FROM:<reverse-path> <CRLF> and
 * two extensions, namely XVERP and SMTPUTF8.
 *
 * The server's EHLO response is not checked. If use of either
 * extensions is enabled even though the server does not support
 * that, mail submission will fail.
 *
 * XVERP is documented at https://www.postfix.org/VERP_README.html
 * and SMTPUTF8 is specified in RFC 6531.
 *
 * @param string $from Source address of this message
 *
 * @return bool
 */
public function mail($from)
{
    $useVerp = ($this->do_verp ? ' XVERP' : '');
    $useSmtputf8 = ($this->do_smtputf8 ? ' SMTPUTF8' : '');

    return $this->sendCommand(
        'MAIL FROM',
        'MAIL FROM:<' . $from . '>' . $useSmtputf8 . $useVerp,
        250
    );
}

/***
 * Send an SMTP QUIT command.
 * Closes the socket if there is no error or the $close_on_error
argument is true.
 * Implements from RFC 821: QUIT <CRLF>.
 *

```

```

        * @param bool $close_on_error Should the connection close if an
error occurs?
        *
        * @return bool
        */
public function quit($close_on_error = true)
{
    $noerror = $this->sendCommand('QUIT', 'QUIT', 221);
    $err = $this->error; //Save any error
    if ($noerror || $close_on_error) {
        $this->close();
        $this->error = $err; //Restore any error from the quit
command
    }

    return $noerror;
}

/**
 * Send an SMTP RCPT command.
 * Sets the TO argument to $toaddr.
 * Returns true if the recipient was accepted false if it was
rejected.
 * Implements from RFC 821: RCPT <SP> TO:<forward-path> <CRLF>.
 *
 * @param string $address The address the message is being sent to
 * @param string $dsn     Comma separated list of DSN notifications.
NEVER, SUCCESS, FAILURE
 *
 * or DELAY. If you specify NEVER all other
notifications are ignored.
 *
 * @return bool
 */
public function recipient($address, $dsn = '')
{
    if (empty($dsn)) {
        $rcpt = 'RCPT TO:<' . $address . '>';
    } else {
        $dsn = strtoupper($dsn);
        $notify = [];

        if (strpos($dsn, 'NEVER') !== false) {
            $notify[] = 'NEVER';
        } else {
            foreach (['SUCCESS', 'FAILURE', 'DELAY'] as $value) {
                if (strpos($dsn, $value) !== false) {
                    $notify[] = $value;
                }
            }
        }
    }
}

```

```

        }
    }

    $rcpt = 'RCPT TO:<' . $address . '> NOTIFY=' . implode(',', 
$notify);
}

return $this->sendCommand(
    'RCPT TO',
    $rcpt,
    [250, 251]
);
}

/***
 * Send SMTP XCLIENT command to server and check its return code.
 *
 * @return bool True on success
 */
public function xclient(array $vars)
{
    $xclient_options = "";
    foreach ($vars as $key => $value) {
        if (in_array($key, SMTP::$xclient_allowed_attributes)) {
            $xclient_options .= " {$key}={$value}";
        }
    }
    if (!$xclient_options) {
        return true;
    }
    return $this->sendCommand('XCLIENT', 'XCLIENT' .
$xclient_options, 250);
}

/***
 * Send an SMTP RSET command.
 * Abort any transaction that is currently in progress.
 * Implements RFC 821: RSET <CRLF>.
 *
 * @return bool True on success
 */
public function reset()
{
    return $this->sendCommand('RSET', 'RSET', 250);
}

/***
 * Send a command to an SMTP server and check its return code.
*/

```

```

*
* @param string    $command      The command name - not sent to the
server
* @param string    $commandstring The actual command to send
* @param int|array $expect       One or more expected integer
success codes
*
* @return bool True on success
*/
protected function sendCommand($command, $commandstring, $expect)
{
    if (!$this->connected()) {
        $this->setError("Called $command without being connected");

        return false;
    }
    //Reject line breaks in all commands
    if ((strpos($commandstring, "\n") !== false) ||
(strpos($commandstring, "\r") !== false)) {
        $this->setError("Command '$command' contained line breaks");

        return false;
    }
    $this->client_send($commandstring . static::LE, $command);

    $this->last_reply = $this->get_lines();
    //Fetch SMTP code and possible error code explanation
    $matches = [];
    if (preg_match('/^([^\d]{3})[ -](?:([\d]\.\.[\d]\.\.[\d]{1,2})) ?/', $this->last_reply, $matches)) {
        $code = (int) $matches[1];
        $code_ex = (count($matches) > 2 ? $matches[2] : null);
        //Cut off error code from each response line
        $detail = preg_replace(
            "/{$code}[ -]" .
            ($code_ex ? str_replace('.', '\.\.', $code_ex) . ' ' : '') . '/m',
            '',
            $this->last_reply
        );
    } else {
        //Fall back to simple parsing if regex fails
        $code = (int) substr($this->last_reply, 0, 3);
        $code_ex = null;
        $detail = substr($this->last_reply, 4);
    }
}

```

```

        $this->edebug('SERVER -> CLIENT: ' . $this->last_reply,
self::DEBUG_SERVER);

        if (!in_array($code, (array) $expect, true)) {
            $this->setError(
                "$command command failed",
                $detail,
                $code,
                $code_ex
            );
            $this->edebug(
                'SMTP ERROR: ' . $this->error['error'] . ': ' . $this-
>last_reply,
                self::DEBUG_CLIENT
            );
            return false;
        }

        //Don't clear the error store when using keepalive
        if ($command !== 'RSET') {
            $this->setError('');
        }

        return true;
    }

    /**
     * Send an SMTP SAML command.
     * Starts a mail transaction from the email address specified in
     $from.
     * Returns true if successful or false otherwise. If True
     * the mail transaction is started and then one or more recipient
     * commands may be called followed by a data command. This command
     * will send the message to the users terminal if they are logged
     * in and send them an email.
     * Implements RFC 821: SAML <SP> FROM:<reverse-path> <CRLF>.
     *
     * @param string $from The address the message is from
     *
     * @return bool
     */
    public function sendAndMail($from)
    {
        return $this->sendCommand('SAML', "SAML FROM:$from", 250);
    }

    /**

```

```

        * Send an SMTP VRFY command.
        *
        * @param string $name The name to verify
        *
        * @return bool
        */
    public function verify($name)
    {
        return $this->sendCommand('VRFY', "VRFY $name", [250, 251]);
    }

    /**
     * Send an SMTP NOOP command.
     * Used to keep keep-alives alive, doesn't actually do anything.
     *
     * @return bool
     */
    public function noop()
    {
        return $this->sendCommand('NOOP', 'NOOP', 250);
    }

    /**
     * Send an SMTP TURN command.
     * This is an optional command for SMTP that this class does not
support.
     * This method is here to make the RFC821 Definition complete for
this class
     * and _may_ be implemented in future.
     * Implements from RFC 821: TURN <CRLF>.
     *
     * @return bool
     */
    public function turn()
    {
        $this->setError('The SMTP TURN command is not implemented');
        $this->edebug('SMTP NOTICE: ' . $this->error['error'],
self::DEBUG_CLIENT);

        return false;
    }

    /**
     * Send raw data to the server.
     *
     * @param string $data    The data to send
     * @param string $command Optionally, the command this is part of,
used only for controlling debug output

```

```

        *
        * @return int|bool The number of bytes sent to the server or false
on error
        */
        public function client_send($data, $command = '')
        {
            //If SMTP transcripts are left enabled, or debug output is posted
online
                //it can leak credentials, so hide credentials in all but lowest
level
            if (
                self::DEBUG_LOWLEVEL > $this->do_debug &&
                in_array($command, ['User & Password', 'Username',
'Password'], true)
            ) {
                $this->edebug('CLIENT -> SERVER: [credentials hidden]',
self::DEBUG_CLIENT);
            } else {
                $this->edebug('CLIENT -> SERVER: ' . $data,
self::DEBUG_CLIENT);
            }
            set_error_handler(function () {
                call_user_func_array([$this, 'errorHandler'],
func_get_args());
            });
            $result = fwrite($this->smtp_conn, $data);
            restore_error_handler();

            return $result;
        }

        /**
         * Get the latest error.
         *
         * @return array
         */
        public function getError()
        {
            return $this->error;
        }

        /**
         * Get SMTP extensions available on the server.
         *
         * @return array|null
         */
        public function getServerExtList()
        {

```

```

        return $this->server_caps;
    }

    /**
     * Get metadata about the SMTP server from its HELO/EHLO response.
     * The method works in three ways, dependent on argument value and
     * current state:
     *   1. HELO/EHLO has not been sent - returns null and populates
     *      $this->error.
     *   2. HELO has been sent -
     *      $name == 'HELO': returns server name
     *      $name == 'EHLO': returns boolean false
     *      $name == any other string: returns null and populates $this-
     *      >error
     *   3. EHLO has been sent -
     *      $name == 'HELO' | 'EHLO': returns the server name
     *      $name == any other string: if extension $name exists, returns
     *      True
     *          or its options (e.g. AUTH mechanisms supported). Otherwise
     *      returns False.
     *
     * @param string $name Name of SMTP extension or 'HELO' | 'EHLO'
     *
     * @return string|bool|null
     */
    public function getServerExt($name)
    {
        if (!$this->server_caps) {
            $this->setError('No HELO/EHLO was sent');

            return null;
        }

        if (!array_key_exists($name, $this->server_caps)) {
            if ('HELO' === $name) {
                return $this->server_caps['EHLO'];
            }
            if ('EHLO' === $name || array_key_exists('EHLO', $this-
>server_caps)) {
                return false;
            }
            $this->setError('HELO handshake was used; No information
about server extensions available');

            return null;
        }

        return $this->server_caps[$name];
    }
}

```

```

}

/***
 * Get the last reply from the server.
 *
 * @return string
 */
public function getLastReply()
{
    return $this->last_reply;
}

/***
 * Read the SMTP server's response.
 * Either before eof or socket timeout occurs on the operation.
 * With SMTP we can tell if we have more lines to read if the
 * 4th character is '-' symbol. If it is a space then we don't
 * need to read anything else.
 *
 * @return string
 */
protected function get_lines()
{
    //If the connection is bad, give up straight away
    if (!is_resource($this->smtp_conn)) {
        return '';
    }
    $data = '';
    $endtime = 0;
    stream_set_timeout($this->smtp_conn, $this->Timeout);
    if ($this->Timelimit > 0) {
        $endtime = time() + $this->Timelimit;
    }
    $selR = [$this->smtp_conn];
    $selW = null;
    while (is_resource($this->smtp_conn) && !feof($this->smtp_conn))
    {
        //Must pass vars in here as params are by reference
        //solution for signals inspired by
https://github.com/symfony/symfony/pull/6540
        set_error_handler(function () {
            call_user_func_array([$this, 'errorHandler'],
func_get_args()));
        });
        $n = stream_select($selR, $selW, $selW, $this->Timelimit);
        restore_error_handler();

        if ($n === false) {

```

```

$message = $this->getError()['detail'];

$this->edebug(
    'SMTP -> get_lines(): select failed (' . $message .
'),',
    self::DEBUG_LOWLEVEL
);

//stream_select returns false when the `select` system
call is interrupted
//by an incoming signal, try the select again
if (stripos($message, 'interrupted system call') !==
false) {
    $this->edebug(
        'SMTP -> get_lines(): retrying stream_select',
        self::DEBUG_LOWLEVEL
    );
    $this->setError('');
    continue;
}

break;
}

if (!$n) {
    $this->edebug(
        'SMTP -> get_lines(): select timed-out in (' . $this-
>Timelimit . ' sec)',
        self::DEBUG_LOWLEVEL
    );
    break;
}

//Deliberate noise suppression - errors are handled
afterwards
$str = @fgets($this->smtp_conn, self::MAX_REPLY_LENGTH);
$this->edebug('SMTP INBOUND: "' . trim($str) . '"',
self::DEBUG_LOWLEVEL);
$data .= $str;
//If response is only 3 chars (not valid, but RFC5321 S4.2
says it must be handled),
//or 4th character is a space or a line break char, we are
done reading, break the loop.
//String array access is a significant micro-optimisation
over strlen
if (!isset($str[3]) || $str[3] === ' ' || $str[3] === "\r" ||
$str[3] === "\n") {
    break;
}

```

```

    }
    //Timed-out? Log and break
    $info = stream_get_meta_data($this->smtp_conn);
    if ($info['timed_out']) {
        $this->edebug(
            'SMTP -> get_lines(): stream timed-out (' . $this-
>Timeout . ' sec)',
            self::DEBUG_LOWLEVEL
        );
        break;
    }
    //Now check if reads took too long
    if ($endtime && time() > $endtime) {
        $this->edebug(
            'SMTP -> get_lines(): timelimit reached (' .
            $this->Timelimit . ' sec)',
            self::DEBUG_LOWLEVEL
        );
        break;
    }
}

return $data;
}

/**
 * Enable or disable VERP address generation.
 *
 * @param bool $enabled
 */
public function setVerp($enabled = false)
{
    $this->do_verp = $enabled;
}

/**
 * Get VERP address generation mode.
 *
 * @return bool
 */
public function getVerp()
{
    return $this->do_verp;
}

/**
 * Enable or disable use of SMTPUTF8.
 *

```

```

        * @param bool $enabled
    */
public function setSMTPUTF8($enabled = false)
{
    $this->do_smtputf8 = $enabled;
}

/**
 * Get SMTPUTF8 use.
 *
 * @return bool
 */
public function getSMTPUTF8()
{
    return $this->do_smtputf8;
}

/**
 * Set error messages and codes.
 *
 * @param string $message      The error message
 * @param string $detail       Further detail on the error
 * @param string $smtp_code    An associated SMTP error code
 * @param string $smtp_code_ex Extended SMTP code
 */
protected function setError($message, $detail = '', $smtp_code = '',
$smtp_code_ex = '')
{
    $this->error = [
        'error' => $message,
        'detail' => $detail,
        'smtp_code' => $smtp_code,
        'smtp_code_ex' => $smtp_code_ex,
    ];
}

/**
 * Set debug output method.
 *
 * @param string|callable $method The name of the mechanism to use
for debugging output, or a callable to handle it
 */
public function setDebugOutput($method = 'echo')
{
    $this->Debugoutput = $method;
}

/**

```

```

        * Get debug output method.
        *
        * @return string
        */
public function getDebugOutput()
{
    return $this->Debugoutput;
}

/**
 * Set debug output level.
 *
 * @param int $level
 */
public function setDebugLevel($level = 0)
{
    $this->do_debug = $level;
}

/**
 * Get debug output level.
 *
 * @return int
 */
public function getDebugLevel()
{
    return $this->do_debug;
}

/**
 * Set SMTP timeout.
 *
 * @param int $timeout The timeout duration in seconds
 */
public function setTimeout($timeout = 0)
{
    $this->Timeout = $timeout;
}

/**
 * Get SMTP timeout.
 *
 * @return int
 */
public function getTimeout()
{
    return $this->Timeout;
}

```

```

/**
 * Reports an error number and string.
 *
 * @param int    $errno    The error number returned by PHP
 * @param string $errmsg  The error message returned by PHP
 * @param string $errfile The file the error occurred in
 * @param int    $errline The line number the error occurred on
 */
protected function errorHandler($errno, $errmsg, $errfile = '',
$errline = 0)
{
    $notice = 'Connection failed.';
    $this->setError(
        $notice,
        $errmsg,
        (string) $errno
    );
    $this->edebug(
        "$notice Error #$errno: $errmsg [$errfile line $errline]",
        self::DEBUG_CONNECTION
    );
}

/**
 * Extract and return the ID of the last SMTP transaction based on
 * a list of patterns provided in
SMTP::$smtp_transaction_id_patterns.
 * Relies on the host providing the ID in response to a DATA command.
 * If no reply has been received yet, it will return null.
 * If no pattern was matched, it will return false.
 *
 * @return bool|string|null
 */
protected function recordLastTransactionID()
{
    $reply = $this->getLastReply();

    if (empty($reply)) {
        $this->last_smtp_transaction_id = null;
    } else {
        $this->last_smtp_transaction_id = false;
        foreach ($this->smtp_transaction_id_patterns as
$smtp_transaction_id_pattern) {
            $matches = [];
            if (preg_match($smtp_transaction_id_pattern, $reply,
$matches)) {
                $this->last_smtp_transaction_id = trim($matches[1]);
            }
        }
    }
}

```

```

                break;
            }
        }

        return $this->last_smtp_transaction_id;
    }

    /**
     * Get the queue/transaction ID of the last SMTP transaction
     * If no reply has been received yet, it will return null.
     * If no pattern was matched, it will return false.
     *
     * @return bool|string|null
     *
     * @see recordLastTransactionID()
     */
    public function getLastTransactionID()
    {
        return $this->last_smtp_transaction_id;
    }
}

```

|   └─ WhatsAppService.php

```

<?php
/**
 * Clase WhatsAppService
 *
 * Servicio para interactuar con una API de WhatsApp a través de
 * peticiones HTTP.
 * Permite verificar el estado del servicio y enviar mensajes.
 *
 * Usa cURL para realizar solicitudes GET y POST, y maneja respuestas en
 * formato JSON.
 */
class WhatsAppService {

    /**
     * @var string $base URL base del servicio de WhatsApp API
     */
    private $base;

    /**
     * Constructor de la clase
     *
     * @param string $base URL base de la API (ejemplo:
     * http://localhost:3000)
     */
}

```

```

public function __construct($base) {
    $this->base = rtrim($base, '/');
}

/**
 * Verifica el estado del servicio WhatsApp API
 *
 * Realiza una petición GET al endpoint `/status-json`.
 *
 * @return array Respuesta de la API decodificada como array asociativo
 * @throws Exception Si ocurre un error en la petición
 */
public function status() {
    return $this->request('GET', '/status-json');
}

/**
 * Envía un mensaje a un número específico
 *
 * Realiza una petición POST al endpoint `/send` .
 *
 * @param string $to Número de teléfono destino en formato válido para
la API
 * @param string $message Contenido del mensaje a enviar
 * @return array Respuesta de la API en formato array asociativo
 * @throws Exception Si ocurre un error en la petición
 */
public function send($to, $message) {
    return $this->request('POST', '/send', ['to' => $to, 'message' =>
$message]);
}

/**
 * Método privado para realizar las solicitudes HTTP a la API
 *
 * Maneja tanto GET como POST. Para POST se envían los datos como JSON.
 *
 * @param string $method Método HTTP a usar ("GET" o "POST")
 * @param string $path Ruta del endpoint de la API (ejemplo: "/send")
 * @param array|null $payload Datos a enviar en el cuerpo de la
petición (solo para POST)
 * @return array Respuesta de la API decodificada en array asociativo
 * @throws Exception Si ocurre un error de conexión o si la API
devuelve un código >= 400
 */
private function request($method, $path, $payload = null) {
    $url = $this->base . $path;
    $ch = curl_init($url);
}

```

```

curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
if ($method === 'POST') {
    curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_HTTPHEADER, ['Content-Type:
application/json']);
    curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($payload,
JSON_UNESCAPED_UNICODE));
}
$res = curl_exec($ch);
if ($res === false) { throw new Exception(curl_error($ch)); }
$code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

$data = json_decode($res, true);
if ($code >= 400) { throw new Exception(($data['error'] ?? 'HTTP
' . $code)); }
return $data ?: [];
}
}

```

```

|
└── model/
    └── Bitacora/
        └── BitacoraDAO.php

```

```

<?php
// Se incluyen los archivos necesarios para mapear la bitácora y la
conexión a la base de datos
require_once 'BitacoraMapper.php';
require_once __DIR__ . '/../../config/Database.php';

/**
 * Clase BitacoraDAO
 *
 * Se encarga de manejar todas las operaciones relacionadas con la tabla
"bitacora".
 * Implementa un patrón DAO (Data Access Object) para mantener separada
la lógica
 * de acceso a datos respecto al resto de la aplicación.
*/
class BitacoraDAO {
    // Conexión a la base de datos
    private $conn;

    /**
     * Constructor
     * Recibe la conexión PDO a la base de datos y la asigna a la clase.
    */

```

```

public function __construct(PDO $db) {
    $this->conn = $db;
}

/**
 * Crear nueva entrada en la bitácora
 * Inserta un registro en la tabla utilizando un procedimiento
almacenado.
 */
public function create(BitacoraDTO $bitacora) {
    $stmt = $this->conn->prepare("CALL BitacoraCreate(?, ?, ?, ?)");
    return $stmt->execute([
        $bitacora->idUsuario,
        $bitacora->horaEntrada,
        $bitacora->horaSalida,
        $bitacora->fecha
    ]);
}

/**
 * Leer una entrada por ID
 * (No se implementa ya que la tabla no tiene una clave primaria
única).
 */
public function read($id) {
    return null; // no se puede implementar sin una PK única
}

/**
 * Leer todas las entradas de la bitácora
 * Incluye también el nombre del usuario mediante un JOIN.
 * Se ordena mostrando primero las sesiones abiertas (sin hora de
salida).
 */
public function readAll() {
    $sql = "SELECT b.*, u.Usuario AS nombreUsuario
            FROM bitacora b
            JOIN usuario u ON b.IdUsuario = u.Id
            ORDER BY
                CASE
                    WHEN b.horaSalida IS NULL OR b.horaSalida =
'00:00:00' THEN 0
                    ELSE 1
                END,
                b.Fecha DESC,
                b.HoraEntrada DESC";
}

$stmt = $this->conn->prepare($sql);

```

```

$stmt->execute();
$result = $stmt->fetchAll(PDO::FETCH_ASSOC);

$bitacoras = [];
foreach ($result as $row) {
    $bitacoras[] = BitacoraMapper::mapRowToDTO($row);
}

return $bitacoras;
}

/**
 * Actualizar la hora de salida
 * Se localiza el registro usando varias condiciones (usuario, fecha
y hora de entrada).
 */
public function update(BitacoraDTO $dto) {
    $sql = "UPDATE bitacora
            SET horaSalida = :horaSalida
            WHERE IdUsuario = :idUsuario
            AND Fecha = :fecha
            AND horaEntrada = :horaEntrada";

    $stmt = $this->conn->prepare($sql);
    $stmt->bindParam(':horaSalida', $dto->horaSalida);
    $stmt->bindParam(':idUsuario', $dto->idUsuario);
    $stmt->bindParam(':fecha', $dto->fecha);
    $stmt->bindParam(':horaEntrada', $dto->horaEntrada);
    return $stmt->execute();
}

/**
 * Leer todas las entradas asociadas a un usuario
 * Se devuelve una lista ordenada por fecha y hora de entrada.
 */
public function readByUsuario($idUsuario) {
    $stmt = $this->conn->prepare("SELECT * FROM bitacora WHERE
IdUsuario = ? ORDER BY Fecha DESC, HoraEntrada DESC");
    $stmt->execute([$idUsuario]);

    $result = [];
    while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
        $result[] = BitacoraMapper::mapRowToDTO($row);
    }
    return $result;
}

/**

```

```

        * Eliminar registros expirados
        * Borra registros cuya hora de salida esté definida y cuya
fecha/hora sea mayor a 30 días.
    */
    public function eliminarExpirados() {
        $sql = "DELETE FROM bitacora
                WHERE (horaSalida IS NOT NULL AND horaSalida != 
'00:00:00')
                        AND CONCAT(fecha, ' ', horaEntrada) < NOW() - INTERVAL 30
DAY";
        $stmt = $this->conn->prepare($sql);
        return $stmt->execute();
    }

    /**
     * Verificar si el usuario ya tiene una sesión activa
     * Retorna true si el usuario tiene una entrada sin hora de salida
registrada.
    */
    public function tieneSesionActiva($idUsuario) {
        $sql = "SELECT COUNT(*) FROM bitacora
                WHERE IdUsuario = ? AND (horaSalida IS NULL OR horaSalida
= '00:00:00')";
        $stmt = $this->conn->prepare($sql);
        $stmt->execute([$idUsuario]);
        return $stmt->fetchColumn() > 0;
    }

    /**
     * Limpieza automática
     * Método estático que crea una nueva conexión y elimina registros
expirados.
    */
    public static function ejecutarLimpiezaAutomatica() {
        $db = (new Database())->getConnection();
        $dao = new BitacoraDAO($db);
        return $dao->eliminarExpirados();
    }
}
?>
```

| | └─ BitacoraDTO.php

```
<?php
/**
 * Clase BitacoraDTO
 *
 * Objeto de Transferencia de Datos (DTO) para la entidad Bitácora.
```

```

* Se utiliza para transportar información entre capas de la aplicación.
* Contiene atributos básicos de la tabla "bitacora" y métodos
* getter/setter para manipularlos.
*/
class BitacoraDTO {
    // Identificador único de la bitácora (si aplica)
    public $id;

    // Relación con el usuario que generó el registro
    public $idUsuario;

    // Hora de entrada del usuario
    public $horaEntrada;

    // Hora de salida del usuario
    public $horaSalida;

    // Fecha del registro
    public $fecha;

    // Nombre del usuario asociado (dato adicional para mostrar en
    vistas)
    public $nombreUsuario; // <-- Nuevo campo

    /**
     * Obtiene el ID del registro de la bitácora
     */
    public function getId() {
        return $this->id;
    }

    /**
     * Asigna el ID al registro de la bitácora
     */
    public function setId($id) {
        $this->id = $id;
    }

    /**
     * Obtiene el ID del usuario
     */
    public function getIdUsuario() {
        return $this->idUsuario;
    }

    /**
     * Asigna el ID del usuario
     */

```

```
public function setIdUsuario($idUsuario) {
    $this->idUsuario = $idUsuario;
}

/**
 * Obtiene la hora de entrada
 */
public function getHoraEntrada() {
    return $this->horaEntrada;
}

/**
 * Asigna la hora de entrada
 */
public function setHoraEntrada($horaEntrada) {
    $this->horaEntrada = $horaEntrada;
}

/**
 * Obtiene la hora de salida
 */
public function getHoraSalida() {
    return $this->horaSalida;
}

/**
 * Asigna la hora de salida
 */
public function setHoraSalida($horaSalida) {
    $this->horaSalida = $horaSalida;
}

/**
 * Obtiene la fecha del registro
 */
public function getFecha() {
    return $this->fecha;
}

/**
 * Asigna la fecha del registro
 */
public function setFecha($fecha) {
    $this->fecha = $fecha;
}

/**
 * Obtiene el nombre del usuario asociado
*/
```

```

        */
    public function getNombreUsuario() {
        return $this->nombreUsuario;
    }

    /**
     * Asigna el nombre del usuario asociado
     */
    public function setNombreUsuario($nombreUsuario) {
        $this->nombreUsuario = $nombreUsuario;
    }
}
?>

```

| | └ BitacoraMapper.php

```

<?php
// Archivo: model/bitacora/BitacoraMapper.php

require_once 'BitacoraDTO.php';

/**
 * Clase BitacoraMapper
 *
 * Su función es transformar las filas obtenidas de la base de datos
 * en objetos de tipo BitacoraDTO. De esta manera, se desacopla
 * la estructura de la base de datos del resto de la aplicación.
 */
class BitacoraMapper
{
    /**
     * Convierte una fila asociativa de la base de datos en un objeto
     BitacoraDTO.
     *
     * Este método recibe un array asociativo (resultado de una consulta)
     * y crea un nuevo objeto BitacoraDTO con los valores
     correspondientes.
     * En caso de que alguna columna no exista, se asigna null o un valor
     vacío.
     */
    public static function mapRowToDTO($row)
    {
        // Creamos una nueva instancia del DTO
        $dto = new BitacoraDTO();

        // Como la tabla no maneja un campo Id único, lo dejamos como
        null
        $dto->id = null;
    }
}

```

```

        // Se asignan los campos desde la fila de la BD,
        // utilizando el operador ?? para manejar valores inexistentes
        $dto->idUsuario      = $row['IdUsuario'] ?? null;
        $dto->horaEntrada    = $row['HoraEntrada'] ?? null;
        $dto->horaSalida     = $row['HoraSalida'] ?? null;
        $dto->fecha          = $row['Fecha'] ?? null;

        // Campo adicional: nombre del usuario asociado (proveniente del
JOIN con usuario)
        $dto->nombreUsuario = $row['nombreUsuario'] ?? ''; // ← Añadido

        // Se devuelve el objeto ya construido
        return $dto;
    }
}

```

```

|   ┌── cliente/
|   |   └── ClienteDAO.php

```

```

<?php
// Archivo: model.cliente/ClienteDAO.php

require_once 'ClienteDTO.php';
require_once 'ClienteMapper.php';

/**
 * Clase ClienteDAO
 *
 * Objeto de Acceso a Datos (DAO) para la entidad Cliente.
 * Se encarga de realizar todas las operaciones de inserción, lectura,
 * actualización y eliminación en la base de datos relacionadas con
clientes.
 * Utiliza procedimientos almacenados para la mayoría de las operaciones
 * y mappers para transformar los resultados en objetos DTO.
 */
class ClienteDAO
{
    // Conexión a la base de datos
    private $conn;

    /**
     * Constructor
     * Recibe una conexión PDO y la asigna a la clase.
     */
    public function __construct($db)
    {
        $this->conn = $db;
    }
}

```

```

}

/**
 * Crear un nuevo cliente
 * Inserta un cliente en la base de datos mediante el procedimiento
almacenado ClienteCreate.
 */
public function create($cliente)
{
    try {
        $stmt = $this->conn->prepare("CALL ClienteCreate(?, ?, ?, ?, ?, ?, ?, ?)");
        return $stmt->execute([
            $cliente->cedula,
            $cliente->nOMBRE,
            $cliente->correo,
            $cliente->telefono,
            $cliente->lugarResidencia,
            $cliente->fechaCumpleanos,
            $cliente->alergias,
            $cliente->gustosEspeciales
        ]);
    } catch (PDOException $e) {
        // Se captura cualquier error PDO y se devuelve en formato
JSON
        echo json_encode([
            'success' => false,
            'message' => 'Error PDO: ' . $e->getMessage()
        ]);
        exit;
    }
}

/**
 * Leer un cliente por su ID
 * Retorna un objeto ClienteDTO o null si no se encuentra.
 */
public function read($id)
{
    try {
        $stmt = $this->conn->prepare("CALL ClienteRead(?)");
        $stmt->execute([$id]);
        $row = $stmt->fetch(PDO::FETCH_ASSOC);
        return $row ? ClienteMapper::mapRowToDTO($row) : null;
    } catch (PDOException $e) {
        error_log("Error al leer cliente: " . $e->getMessage());
        return null;
    }
}

```

```

}

/**
 * Leer todos los clientes
 * Retorna una lista de ClienteDTO con los clientes obtenidos.
 * Se calcula también el total histórico de compras de cada cliente.
 */
public function readAll()
{
    try {
        $stmt = $this->conn->prepare("CALL ClienteReadAll()");
        $stmt->execute();
        $clientes = [];
        while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
            // Obtener total histórico manualmente por cliente
            $row['TotalHistorico'] = $this-
>obtenerTotalHistoricoPorCliente($row['Id']);
            $clientes[] = ClienteMapper::mapRowToDTO($row);
        }
        return $clientes;
    } catch (PDOException $e) {
        error_log("Error al leer todos los clientes: " . $e-
>getMessage());
        return [];
    }
}

/**
 * Obtener el total histórico de compras de un cliente
 * Consulta la tabla de compra y suma los montos por cliente.
 */
private function obtenerTotalHistoricoPorCliente($idCliente)
{
    try {
        $stmt = $this->conn->prepare("SELECT IFNULL(SUM(Monto), 0) AS
Total FROM compra WHERE IdCliente = ?");
        $stmt->execute([$idCliente]);
        $result = $stmt->fetch(PDO::FETCH_ASSOC);
        return $result ? $result['Total'] : 0;
    } catch (PDOException $e) {
        error_log("Error al obtener total histórico: " . $e-
>getMessage());
        return 0;
    }
}

/**

```

```

        * Verifica si ya existe un teléfono o correo registrado en otro
cliente
        * Se utiliza un procedimiento almacenado para detectar duplicados.
        */
public function existeTelefonoOCorreo($telefono, $correo, $id = null)
{
    try {
        $stmt = $this->conn->prepare("CALL
ClienteVerificarDuplicados(?, ?, ?)");
        $stmt->execute([$telefono, $correo, $id]);
        return $stmt->fetch(PDO::FETCH_ASSOC) ? true : false;
    } catch (PDOException $e) {
        error_log("Error al verificar duplicados (SP): " . $e-
>getMessage());
        return true;
    }
}

/**
 * Verifica si ya existe una cédula registrada en otro cliente
 * Se usa un procedimiento almacenado para la verificación.
 */
public function existeCedula($cedula, $id = null) {
    try {
        $stmt = $this->conn->prepare("CALL ClienteExisteCedula(?, ?)");
        $stmt->execute([$cedula, $id]);
        return $stmt->fetch(PDO::FETCH_ASSOC) ? true : false;
    } catch (PDOException $e) {
        error_log("Error al verificar cédula duplicada: " . $e-
>getMessage());
        return true;
    }
}

/**
 * Actualizar un cliente existente
 * Modifica los datos de un cliente en la base de datos con el
procedimiento almacenado ClienteUpdate.
*/
public function update($cliente)
{
    try {
        $stmt = $this->conn->prepare("CALL ClienteUpdate(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
        return $stmt->execute([
            $cliente->id,
            $cliente->cedula,

```

```

        $cliente->nombre,
        $cliente->correo,
        $cliente->telefono,
        $cliente->lugarResidencia,
        $cliente->fechaCumpleanos,
        $cliente->acumulado,
        $cliente->alergias,
        $cliente->gustosEspeciales
    ]);
} catch (PDOException $e) {
    error_log("Error al actualizar cliente: " . $e-
>getMessage());
    return false;
}
}

/**
 * Eliminar un cliente por su ID
 * Utiliza el procedimiento almacenado ClienteDelete.
 */
public function delete($id)
{
    try {
        $stmt = $this->conn->prepare("CALL ClienteDelete(?)");
        return $stmt->execute([$id]);
    } catch (PDOException $e) {
        error_log("Error al eliminar cliente: " . $e->getMessage());
        return false;
    }
}
}

```

### | | └─ ClienteDTO.php

```

<?php
// Archivo: model/cliente/ClienteDTO.php

/**
 * Clase ClienteDTO
 *
 * Objeto de Transferencia de Datos (DTO) para la entidad Cliente.
 * Se utiliza para transportar información de clientes entre
 * la base de datos y las distintas capas de la aplicación.
 *
 * Contiene únicamente atributos públicos, sin lógica adicional,
 * lo que permite que los mappers y DAOs asignen valores directamente.
 */
class ClienteDTO

```

```

{
    // Identificador único del cliente en la base de datos
    public $id;

    // Número de cédula del cliente
    public $cedula;

    // Nombre completo del cliente
    public $nombre;

    // Correo electrónico del cliente
    public $correo;

    // Número de teléfono del cliente
    public $telefono;

    // Lugar de residencia (domicilio)
    public $lugarResidencia;

    // Fecha de cumpleaños del cliente
    public $fechaCumpleanos;

    // Acumulado de compras u otros valores asociados
    public $acumulado;

    // Fecha en que se registró el cliente en el sistema
    public $fechaRegistro;

    // Total histórico de compras del cliente (dato calculado) ← NUEVO
    public $totalHistorico;

    // Información sobre alergias del cliente ← NUEVO
    public $alergias;

    // Preferencias o gustos especiales del cliente ← NUEVO
    public $gustosEspeciales;
}

```

### | | └ ClienteMapper.php

```

<?php
// Archivo: model/cliente/ClienteMapper.php
require_once 'ClienteDTO.php';

/**
 * Clase ClienteMapper
 *

```

```

    * Se encarga de transformar filas de resultados obtenidos de la base de
    datos
    * en objetos ClienteDTO. De esta manera se desacopla la estructura de la
    BD
    * del resto de la aplicación, facilitando el transporte de datos.
    */
class ClienteMapper
{
    /**
     * Convierte una fila asociativa de la base de datos en un objeto
     ClienteDTO.
     *
     * @param array $row Fila obtenida de la base de datos (asociativa).
     * @return ClienteDTO Objeto DTO con los datos mapeados.
     */
    public static function mapRowToDTO($row)
    {
        // Se crea una nueva instancia del DTO
        $dto = new ClienteDTO();

        // Asignación de campos principales
        $dto->id          = $row['Id'];
        $dto->cedula      = $row['Cedula'];
        $dto->nombre       = $row['Nombre'];
        $dto->correo       = $row['Correo'];
        $dto->telefono     = $row['Telefono'];
        $dto->lugarResidencia = $row['LugarResidencia'];
        $dto->fechaCumpleanos = $row['FechaCumpleanos'];
        $dto->acumulado    = $row['Acumulado'];
        $dto->fechaRegistro = $row['FechaRegistro'];

        // Campos adicionales añadidos al DTO
        $dto->alergias      = $row['Alergias']; // <-
NUEVO
        $dto->gustosEspeciales = $row['GustosEspeciales']; // <-
NUEVO
        $dto->totalHistorico   = $row['TotalHistorico']; // <-
EXISTENTE

        // Se devuelve el objeto DTO completo
        return $dto;
    }
}

```

|   |   └-- codigo/  
 |   |   └-- CódigoDAO.php

<?php

```

// Archivo: model/codigo/CodigoDAO.php

require_once 'CodigoDTO.php';
require_once 'CodigoMapper.php';

/**
 * Clase CodigoDAO
 *
 * Objeto de Acceso a Datos (DAO) para la entidad Código.
 * Se encarga de ejecutar operaciones en la base de datos utilizando
 * procedimientos almacenados para crear, leer, actualizar, eliminar
 * y reasignar códigos de barras asociados a clientes.
 */
class CodigoDAO
{
    // Conexión a la base de datos
    private $conn;

    /**
     * Constructor
     * Recibe una conexión PDO y la asigna a la clase.
     */
    public function __construct($db)
    {
        $this->conn = $db;
    }

    /**
     * Crear un nuevo código de barras
     * Inserta un registro usando el procedimiento almacenado
     CodigoCreate.
     */
    public function create($codigo)
    {
        try {
            $stmt = $this->conn->prepare("CALL CodigoCreate(?, ?, ?)");
            $success = $stmt->execute([
                $codigo->idCliente,
                $codigo->codigoBarra,
                $codigo->cantImpresiones
            ]);
            if (!$success) {
                $errorInfo = $stmt->errorInfo();
                return [
                    'success' => false,
                    'message' => 'Error al crear código: ' .
$errorInfo[2]
                ];
            }
        }
    }
}

```

```

        }
        return [ 'success' => true];
    } catch (PDOException $e) {
        return [
            'success' => false,
            'message' => 'Error PDO (create): ' . $e->getMessage()
        ];
    }
}

/**
 * Leer un código por su ID
 * Devuelve un DTO con los datos o un mensaje de error si no se
encuentra.
 */
public function read($id)
{
    try {
        $stmt = $this->conn->prepare("CALL CodigoRead(?)");
        $stmt->execute([$id]);
        $row = $stmt->fetch(PDO::FETCH_ASSOC);
        if (!$row) {
            return [
                'success' => false,
                'message' => "No se encontró el código con id $id"
            ];
        }
        return [
            'success' => true,
            'data' => CodigoMapper::mapRowToDTO($row)
        ];
    } catch (PDOException $e) {
        return [
            'success' => false,
            'message' => 'Error PDO (read): ' . $e->getMessage()
        ];
    }
}

/**
 * Leer todos los códigos
 * Ejecuta el procedimiento almacenado CodigoReadAll y devuelve la
lista completa.
 */
public function readAll()
{
    try {
        $stmt = $this->conn->prepare("CALL CodigoReadAll()");

```

```

        $stmt->execute();
        $codigos = [];
        while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
            $codigos[] = CódigoMapper::mapRowToDTO($row);
        }
        return [
            'success' => true,
            'data' => $codigos
        ];
    } catch (PDOException $e) {
        return [
            'success' => false,
            'message' => 'Error PDO (readAll): ' . $e->getMessage()
        ];
    }
}

/**
 * Actualizar un código existente
 * Modifica los datos mediante el procedimiento almacenado
CódigoUpdate.
*/
public function update($codigo)
{
    try {
        $stmt = $this->conn->prepare("CALL CódigoUpdate(?, ?, ?)");
        $success = $stmt->execute([
            $codigo->id,
            $codigo->codigoBarra,
            $codigo->cantImpresiones
        ]);
        if (!$success) {
            $errorInfo = $stmt->errorInfo();
            return [
                'success' => false,
                'message' => 'Error al actualizar código: ' .
$errorInfo[2]
            ];
        }
        return ['success' => true];
    } catch (PDOException $e) {
        return [
            'success' => false,
            'message' => 'Error PDO (update): ' . $e->getMessage()
        ];
    }
}

```

```

/**
 * Eliminar un código por su ID
 * Ejecuta el procedimiento almacenado CódigoDelete.
 */
public function delete($id)
{
    try {
        $stmt = $this->conn->prepare("CALL CódigoDelete(?)");
        $success = $stmt->execute([$id]);
        if (!$success) {
            $errorInfo = $stmt->errorInfo();
            return [
                'success' => false,
                'message' => 'Error al eliminar código: ' .
$errorInfo[2]
            ];
        }
        return ['success' => true];
    } catch (PDOException $e) {
        return [
            'success' => false,
            'message' => 'Error PDO (delete): ' . $e->getMessage()
        ];
    }
}

/**
 * Leer un código asociado a un cliente
 * Busca mediante el procedimiento almacenado CódigoSelectByCliente.
 */
public function readByCliente($idCliente)
{
    try {
        $stmt = $this->conn->prepare("CALL
CódigoSelectByCliente(?)");
        $stmt->execute([$idCliente]);
        $row = $stmt->fetch(PDO::FETCH_ASSOC);
        if (!$row) {
            return [
                'success' => false,
                'message' => "No se encontró código activo para el
cliente $idCliente"
            ];
        }
        return [
            'success' => true,
            'data' => CódigoMapper::mapRowToDTO($row)
        ];
    }
}

```

```

    } catch (PDOException $e) {
        return [
            'success' => false,
            'message' => 'Error PDO (readByCliente): ' . $e-
>getMessage()
        ];
    }
}

/**
 * Reasignar un código de barras
 * Llama al procedimiento almacenado CódigoReassign,
 * el cual genera internamente un nuevo código para el cliente.
 */
public function reassignCode($idCliente, $motivo)
{
    try {
        // El procedimiento se encarga de localizar el código activo
y reasignarlo
        $stmt = $this->conn->prepare("CALL CódigoReassign(?, ?)");
        $result = $stmt->execute([$idCliente, $motivo]);

        if ($result) {
            // Obtener el nuevo código generado desde la respuesta
            $nuevoCodigoResult = $stmt->fetch(PDO::FETCH_ASSOC);
            $nuevoCodigo = $nuevoCodigoResult['NuevoCodigoGenerado']
?? '';
        }

        return [
            'success' => true,
            'nuevoCodigo' => $nuevoCodigo,
            'message' => 'Código reasignado exitosamente'
        ];
    } else {
        $errorInfo = $stmt->errorInfo();
        return [
            'success' => false,
            'message' => 'Error al reasignar código: ' .
$errorInfo[2]
        ];
    }
} catch (PDOException $e) {
    return [
        'success' => false,
        'message' => 'Error PDO (reassignCode): ' . $e-
>getMessage()
    ];
}

```

```
    }
}
```

### | | └-- CódigoDTO.php

```
<?php
// Archivo: model/codigo/CódigoDTO.php

/**
 * Clase CódigoDTO
 *
 * Objeto de Transferencia de Datos (DTO) para la entidad Código.
 * Contiene únicamente atributos públicos que representan los campos
 * relacionados con los códigos de barras asignados a clientes.
 *
 * Se utiliza para transportar información entre la base de datos
 * y las diferentes capas de la aplicación sin lógica adicional.
 */
class CódigoDTO
{
    // Identificador único del código en la base de datos
    public $id;

    // Relación con el cliente (número de tarjeta asignada)
    public $idCliente;

    // Valor del código de barras generado
    public $códigoBarra;

    // Cantidad de impresiones realizadas para este código
    public $cantImpresiones;

    // Estado actual del código (ej. Activo / Inactivo)
    public $estado;

    // Motivo registrado cuando el código cambia de estado
    public $motivoCambio;

    // Fecha en que se realizó el cambio de estado
    public $fechaCambio;
}
```

### | | └-- CódigoMapper.php

```
<?php
// Archivo: model/codigo/CódigoMapper.php

require_once 'CódigoDTO.php';
```

```

/**
 * Clase CodigoMapper
 *
 * Se encarga de convertir filas obtenidas desde la base de datos
 * en objetos CodigoDTO. De esta forma se separa la estructura
 * de la BD de la lógica de la aplicación.
 */
class CodigoMapper
{
    /**
     * Convierte una fila de base de datos (array asociativo) en un
     * objeto CodigoDTO.
     *
     * @param array $row Fila de datos proveniente de la consulta SQL.
     * @return CodigoDTO Objeto DTO con los datos mapeados.
     */
    public static function mapRowToDTO($row)
    {
        // Se crea una nueva instancia del DTO
        $dto = new CodigoDTO();

        // Asignación de campos básicos desde la BD
        $dto->id          = $row['Id'];
        $dto->idCliente   = $row['IdCliente'];
        $dto->codigoBarra  = $row['CodigoBarra'];
        $dto->cantImpresiones= $row['CantImpresiones'];

        // Asignación de campos de control del código
        $dto->estado      = $row['Estado'] ?? 'Activo'; // Valor por
defecto "Activo" si no se especifica
        $dto->motivoCambio = $row['MotivoCambio'];
        $dto->fechaCambio  = $row['FechaCambio'];

        // Se devuelve el DTO ya construido
        return $dto;
    }
}

```

```

|   |   └ compra/
|   |       └ CompraDAO.php

```

```

<?php
// Archivo: model/compra/CompraDAO.php

require_once 'CompraDTO.php';
require_once 'CompraMapper.php';

```

```

/**
 * Clase CompraDAO
 *
 * Objeto de Acceso a Datos (DAO) para la entidad Compra.
 * Maneja las operaciones CRUD relacionadas con las compras en la base de
datos,
 * utilizando procedimientos almacenados (SP).
 */
class CompraDAO
{
    // Conexión a la base de datos
    private $conn;

    /**
     * Constructor
     * Recibe una conexión PDO y la asigna a la clase.
     */
    public function __construct($db)
    {
        $this->conn = $db;
    }

    /**
     * Crear una compra
     * Llama al SP "SumarCompra", que se encarga de insertar la compra
     * y actualizar el acumulado del cliente.
     */
    public function create($compra)
    {
        try {
            // 1. Llamar al SP que hace todo (INSERT + UPDATE en
acumulado)
            $stmt = $this->conn->prepare("CALL SumarCompra(?, ?)");
            return $stmt->execute([
                $compra->idCliente,
                $compra->total
            ]);
        } catch (PDOException $e) {
            error_log("Error al registrar compra: " . $e->getMessage());
            return false;
        }
    }

    /**
     * Leer una compra por su ID
     * Retorna un objeto CompraDTO o null si no existe.
     */
    public function read($idCompra)

```

```

{
    try {
        $stmt = $this->conn->prepare("CALL CompraRead(?)");
        $stmt->execute([$idCompra]);
        $row = $stmt->fetch(PDO::FETCH_ASSOC);
        return $row ? CompraMapper::mapRowToDTO($row) : null;
    } catch (PDOException $e) {
        error_log("Error al leer compra: " . $e->getMessage());
        return null;
    }
}

/**
 * Eliminar una compra por su ID
 * Ejecuta el procedimiento almacenado CompraDelete.
 */
public function delete($idCompra)
{
    try {
        $stmt = $this->conn->prepare("CALL CompraDelete(?)");
        return $stmt->execute([$idCompra]);
    } catch (PDOException $e) {
        error_log("Error al eliminar compra: " . $e->getMessage());
        return false;
    }
}

/**
 * Leer todas las compras
 * Llama al SP CompraReadAll y devuelve un array de CompraDTO.
 */
public function readAll()
{
    try {
        $stmt = $this->conn->prepare("CALL CompraReadAll()");
        $stmt->execute();
        $compras = [];
        while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
            $compras[] = CompraMapper::mapRowToDTO($row);
        }
        return $compras;
    } catch (PDOException $e) {
        error_log("Error al leer todas las compras: " . $e-
>getMessage());
        return [];
    }
}

```

```

/**
 * Leer todas las compras de un cliente específico
 * Usa el SP CompraReadByCliente para devolver la lista filtrada.
 */
public function readByCliente($idCliente)
{
    try {
        $stmt = $this->conn->prepare("CALL CompraReadByCliente(?)");
        $stmt->execute([$idCliente]);
        $compras = [];
        while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
            $compras[] = CompraMapper::mapRowToDTO($row);
        }
        return $compras;
    } catch (PDOException $e) {
        error_log("Error al leer compras por cliente: " . $e-
>getMessage());
        return [];
    }
}

```

### | | └─ CompraDTO.php

```

<?php
// Archivo: model/compra/CompraDTO.php

/**
 * Clase CompraDTO
 *
 * Objeto de Transferencia de Datos (DTO) para la entidad Compra.
 * Representa una fila de la tabla 'compra' y se utiliza para
 * transportar información entre la base de datos y las capas
 * de la aplicación sin lógica adicional.
 */
class CompraDTO
{
    /**
     * ID de la compra (clave primaria, autoincremental)
     * Identifica de forma única cada compra registrada.
     */
    public $idCompra;

    /**
     * Fecha en que se realizó la compra
     * Usualmente en formato 'YYYY-MM-DD'.
     */
    public $fechaCompra;
}

```

```

    /**
     * Monto total de la compra
     * Representa el valor económico asociado a la transacción.
     */
    public $total;

    /**
     * ID del cliente asociado a la compra (clave foránea)
     * Permite relacionar la compra con un cliente en específico.
     */
    public $idCliente;
}

```

| | └ CompraMapper.php

```

<?php
// Archivo: model/compra/CompraMapper.php

require_once 'CompraDTO.php';

/**
 * Clase CompraMapper
 *
 * Se encarga de transformar las filas devueltas por la base de datos
 * en objetos CompraDTO. De esta forma se separa la estructura interna
 * de la tabla de la lógica de la aplicación.
 */
class CompraMapper
{
    /**
     * Convierte una fila asociativa de la base de datos en un objeto
     CompraDTO.
     *
     * @param array $row Fila asociativa obtenida de la consulta SQL.
     * @return CompraDTO Objeto DTO con los valores mapeados.
     */
    public static function mapRowToDTO($row)
    {
        // Crear una nueva instancia del DTO
        $dto = new CompraDTO();

        // Asignar valores de la fila a las propiedades del DTO
        $dto->idCompra      = $row['IdCompra'];
        $dto->fechaCompra   = $row['FechaCompra'];
        $dto->total          = $row['Total'];
        $dto->idCliente     = $row['IdCliente'];
    }
}

```

```

        // Retornar el objeto DTO ya construido
        return $dto;
    }
}

```

```

|   └─ cumple/
|       └─ CumpleDAO.php

```

```

<?php
require_once 'CumpleDTO.php';
require_once 'CumpleMapper.php';

/**
 * Clase CumpleDAO
 *
 * Objeto de Acceso a Datos (DAO) para la entidad Cumple.
 * Gestiona operaciones relacionadas con el registro y seguimiento
 * de cumpleaños de clientes, así como su historial y estados.
 */
class CumpleDAO
{
    // Conexión a la base de datos
    private $conn;

    /**
     * Constructor
     * Recibe una conexión PDO y la asigna a la clase.
     */
    public function __construct($db)
    {
        $this->conn = $db;
    }

    /**
     * Obtener cumpleaños de la semana
     *
     * Llama al SP `ClienteCumpleSemanaOffset`, que devuelve los clientes
     * que cumplen años en la semana actual o en la siguiente (según
     * offset).
     *
     * @param int $offset 0 = semana actual, 1 = semana siguiente
     * @return CumpleDTO[] Lista de objetos DTO con los cumpleaños
     * encontrados
     */
    public function obtenerCumplidosSemana(int $offset = 0)
    {
        try {
            // 0 = semana actual, 1 = semana siguiente

```

```

        $stmt = $this->conn->prepare("CALL
ClienteCumpleSemanaOffset(:o)");
        $stmt->bindValue(':o', $offset, PDO::PARAM_INT);
        $stmt->execute();

        $clientes = [];
        while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
            $clientes[] = CumpleMapper::mapRowToDTO($row);
        }
        $stmt->closeCursor();

        // (Opcional) ordenar por fecha de cumpleaños
        usort($clientes, function ($a, $b) {
            $fechaA = new DateTime($a->fechaCumpleanos);
            $fechaB = new DateTime($b->fechaCumpleanos);
            return $fechaA <=> $fechaB;
        });

        return $clientes;
    } catch (PDOException $e) {
        error_log("Error al obtener cumpleaños de la semana: " . $e-
>getMessage());
        return [];
    }
}

/**
 * Actualizar el estado de un cumpleaños
 *
 * Nota: El campo 'estado' ya no existe en la tabla cliente,
 * por lo tanto este método no realiza ninguna acción.
 * Se mantiene por compatibilidad.
 */
public function actualizarEstado($id, $estado)
{
    return true;
}

/**
 * Obtener historial de cumpleaños
 *
 * Devuelve el historial de cumpleaños registrados en la tabla
`cumple`,
 * realizando antes:
 * - Actualización de vencidos (marca como vencidos los registros
pasados).
 * - Purga de registros con más de 30 días.
 *

```

```

    * @return CumpleDTO[] Lista de objetos DTO con el historial
    */
public function obtenerHistorial()
{
    try {
        // ☑ Actualiza los vencidos antes de consultar
        $this->marcarCumpliesVencidos();

        // ☑ Purga los registros con más de 30 días
        $this->purgarHistorial30Dias();

        // ☑ Vuelve a actualizar los vencidos (seguridad extra)
        $this->marcarCumpliesVencidos();

        $sql = "SELECT c.Id, cl.Cedula, cl.Nombre, cl.Correo,
cl.Telefono, cl.FechaCumpleanos,
                    c.FechaLlamada, c.Vence, c.Vencido
        FROM cumple c
        INNER JOIN cliente cl ON c.IdCliente = cl.Id
        ORDER BY FechaLlamada DESC";

        $stmt = $this->conn->prepare($sql);
        $stmt->execute();

        $historial = [];
        while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
            $historial[] = CumpleMapper::mapRowToDTO($row);
        }

        return $historial;
    } catch (PDOException $e) {
        error_log("Error al obtener historial: " . $e->getMessage());
        return [];
    }
}

/**
 * Marcar cumpleaños vencidos
 *
 * Actualiza el campo `Vencido` en los registros cuya fecha límite ya
pasó.
 */
private function marcarCumpliesVencidos()
{
    try {
        $sql = "UPDATE cumple
                SET Vencido = 'SI'
                WHERE Vencido = 'NO' AND Vence < CURDATE()";

```

```

        $this->conn->exec($sql);
    } catch (PDOException $e) {
        error_log("Error al marcar vencidos: " . $e->getMessage());
    }
}

/**
 * Purgar historial con más de 30 días
 *
 * Llama al SP `PurgarHistorialCumple30Dias` para eliminar
 * registros antiguos y mantener la tabla optimizada.
 */
private function purgarHistorial30Dias()
{
    try {
        $stmt = $this->conn->prepare("CALL
PurgarHistorialCumple30Dias()");
        $stmt->execute();
    } catch (PDOException $e) {
        error_log("Error al purgar historial: " . $e->getMessage());
    }
}
}

```

| |   |— CumpleDTO.php

```

<?php

/**
 * Clase CumpleDTO
 *
 * Objeto de Transferencia de Datos (DTO) para la gestión de cumpleaños.
 * Se utiliza para transportar información relacionada con clientes que
cumplen años,
 * permitiendo manejar recordatorios, envíos de correo, llamadas o
estados de atención.
 */
class CumpleDTO
{
    // Identificador único del registro de cumpleaños
    public $id;

    // Cédula del cliente
    public $cedula;

    // Nombre completo del cliente
    public $nombre;
}

```

```

    // Correo electrónico del cliente
    public $correo;

    // Número de teléfono del cliente
    public $telefono;

    // Fecha de cumpleaños del cliente
    public $fechaCumpleanos;

    // Estado actual del cumpleaños (por defecto "PENDIENTE")
    public $estado = 'PENDIENTE'; // Por defecto

    // Fecha en la que se realizó una llamada de seguimiento
    public $fechaLlamada;

    // Fecha límite o de vencimiento de la gestión
    public $vence;

    // Indica si el registro ya está vencido (true/false)
    public $vencido;
}

```

```

| | └─ CumpleMapper.php
<?php

require_once 'CumpleDTO.php';

/**
 * Clase CumpleMapper
 *
 * Se encarga de transformar filas devueltas por la base de datos
 * en objetos CumpleDTO. Esto permite desacoplar la estructura
 * interna de la tabla de cumpleaños respecto al resto de la aplicación.
 */
class CumpleMapper
{
    /**
     * Convierte una fila asociativa de la base de datos en un objeto
     CumpleDTO.
     *
     * @param array $row Fila de datos devuelta por una consulta SQL.
     * @return CumpleDTO Objeto DTO con los valores mapeados.
     */
    public static function mapRowToDTO($row)
    {
        // Crear nueva instancia del DTO

```

```

    $dto = new CumpleDTO();

    // Asignación de valores básicos
    $dto->id          = $row['Id'] ?? null;
    $dto->cedula      = $row['Cedula'] ?? '';
    $dto->nombre       = $row['Nombre'] ?? '';
    $dto->correo       = $row['Correo'] ?? '';
    $dto->telefono     = $row['Telefono'] ?? '';
    $dto->fechaCumpleanos= $row['FechaCumpleanos'] ?? null;

    // Asignación de campos de control del cumpleaños
    $dto->estado       = $row['Estado'] ?? 'Activo';
    $dto->fechaLlamada = $row['FechaLlamada'] ?? null;
    $dto->vence         = $row['Vence'] ?? null;
    $dto->vencido       = $row['Vencido'] ?? null;

    // Devolver objeto DTO listo para uso en la aplicación
    return $dto;
}

}

```

| ┌── tarea/  
 | └── TareaDAO.php

```

<?php
// Archivo: model.tarea/TareaDAO.php

require_once 'TareaDTO.php';
require_once 'TareaMapper.php';

/**
 * Clase TareaDAO
 *
 * Objeto de Acceso a Datos (DAO) para la entidad Tarea.
 * Encargada de gestionar las operaciones CRUD sobre la base de datos
 * mediante procedimientos almacenados (SP).
 */
class TareaDAO
{
    // Conexión a la base de datos
    private $conn;

    /**
     * Constructor
     * Recibe una conexión PDO y la asigna a la clase.
     */
    public function __construct($db)
    {

```

```

        $this->conn = $db;
    }

    /**
     * Crear una nueva tarea
     * Llama al procedimiento almacenado TareaCreate.
     */
    public function create($tarea)
    {
        $stmt = $this->conn->prepare("CALL TareaCreate(?, ?)");
        return $stmt->execute([
            $tarea->descripcion,
            $tarea->estado
        ]);
    }

    /**
     * Leer una tarea por su ID
     * Devuelve un objeto TareaDTO o null si no existe.
     */
    public function read($id)
    {
        $stmt = $this->conn->prepare("CALL TareaRead(?)");
        $stmt->execute([$id]);
        $row = $stmt->fetch(PDO::FETCH_ASSOC);
        return $row ? TareaMapper::mapRowToDTO($row) : null;
    }

    /**
     * Leer todas las tareas
     * Llama al procedimiento almacenado TareaReadAll
     * y devuelve una lista de objetos TareaDTO.
     */
    public function readAll()
    {
        $stmt = $this->conn->prepare("CALL TareaReadAll()");
        $stmt->execute();
        $tareas = [];
        while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
            $tareas[] = TareaMapper::mapRowToDTO($row);
        }
        return $tareas;
    }

    /**
     * Actualizar el estado de una tarea
     * Llama al procedimiento almacenado TareaUpdate.
     */

```

```

public function update($id, $estado)
{
    $stmt = $this->conn->prepare("CALL TareaUpdate(?, ?)");
    return $stmt->execute([$id, $estado]);
}

/**
 * Eliminar una tarea por su ID
 * Llama al procedimiento almacenado TareaDelete.
 */
public function delete($id)
{
    $stmt = $this->conn->prepare("CALL TareaDelete(?)");
    return $stmt->execute([$id]);
}

```

| | └─ TareaDTO.php

```

<?php
// Archivo: model.tarea/TareaDTO.php

/**
 * Clase TareaDTO
 *
 * Objeto de Transferencia de Datos (DTO) para la entidad Tarea.
 * Representa una fila de la tabla `tarea` y se utiliza para transportar
 * información entre la base de datos y las distintas capas de la
 * aplicación.
 */
class TareaDTO
{
    // Identificador único de la tarea (clave primaria)
    public $id;

    // Descripción o detalle de la tarea
    public $descripcion;

    // Estado actual de la tarea (ej. PENDIENTE, COMPLETADA)
    public $estado;

    // Fecha en que la tarea fue creada (asignada automáticamente en BD)
    public $fechaCreacion;
}

```

| | └─ TareaMapper.php

```
<?php
```

```

// Archivo: model.tarea/TareaMapper.php
require_once 'TareaDTO.php';

/**
 * Clase TareaMapper
 *
 * Se encarga de transformar filas obtenidas de la base de datos
 * en objetos TareaDTO. Permite desacoplar la estructura de la tabla
 * de la lógica de negocio de la aplicación.
 */
class TareaMapper
{
    /**
     * Convierte una fila asociativa de la base de datos en un objeto
     * TareaDTO.
     *
     * @param array $row Fila asociativa devuelta por la consulta SQL.
     * @return TareaDTO Objeto DTO con los datos mapeados.
     */
    public static function mapRowToDTO($row)
    {
        // Crear nueva instancia del DTO
        $dto = new TareaDTO();

        // Asignar valores de la fila a las propiedades del DTO
        $dto->id          = $row['id'];
        $dto->descripcion  = $row['descripcion'];
        $dto->estado       = $row['estado'];
        $dto->fechaCreacion = $row['fecha_creacion'];

        // Retornar objeto DTO listo para uso
        return $dto;
    }
}

```

```

|   └─ usuario/
|       └─ UsuarioDAO.php

```

```

<?php
// Archivo: model.usuario/UsuarioDAO.php
require_once 'UsuarioDTO.php';
require_once 'UsuarioMapper.php';

/**
 * Clase UsuarioDAO
 *
 * Objeto de Acceso a Datos (DAO) para la entidad Usuario.
 * Gestiona las operaciones CRUD sobre la base de datos utilizando

```

```

 * procedimientos almacenados (SP).
 */
class UsuarioDAO
{
    // Conexión a la base de datos
    private $conn;

    /**
     * Constructor
     * @param PDO $db Conexión PDO activa
     */
    public function __construct($db)
    {
        $this->conn = $db;
    }

    /**
     * Obtener todos los usuarios
     *
     * @param bool $omitPassword True (por defecto) para ocultar el hash
     * de contraseña,
     *                      False para incluirlo (ej. login).
     * @return UsuarioDTO[] Lista de usuarios
     */
    public function readAll($omitPassword = true)
    {
        try {
            $stmt = $this->conn->prepare("CALL UsuarioReadAll()");
            $stmt->execute();
            $usuarios = [];
            while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
                $usuarios[] = UsuarioMapper::mapRowToDTO($row,
$omitPassword);
            }
            return $usuarios;
        } catch (PDOException $e) {
            error_log("Error al leer todos los usuarios: " . $e-
>getMessage());
            return [];
        }
    }

    /**
     * Crear un nuevo usuario
     *
     * @param UsuarioDTO $usuario Objeto con los datos del usuario
     * @return bool|string True si se creó, mensaje de error si falla
     */

```

```

public function create($usuario)
{
    try {
        $stmt = $this->conn->prepare("CALL UsuarioCreate(?, ?, ?)");
        return $stmt->execute([
            $usuario->usuario,
            $usuario->contrasena,
            $usuario->rol
        ]);
    } catch (PDOException $e) {
        return "Error PDO: " . $e->getMessage();
    }
}

/**
 * Eliminar un usuario por su ID
 */
public function delete($id)
{
    try {
        $stmt = $this->conn->prepare("CALL UsuarioDelete(?)");
        return $stmt->execute([$id]);
    } catch (PDOException $e) {
        error_log("Error al eliminar usuario: " . $e->getMessage());
        return false;
    }
}

/**
 * Obtener un usuario por su ID
 *
 * @param int $id Identificador del usuario
 * @param bool $omitPassword True para no incluir contraseña
 * @return UsuarioDTO|NULL Usuario encontrado o null si no existe
 */
public function read($id, $omitPassword = true)
{
    try {
        $stmt = $this->conn->prepare("CALL UsuarioRead(?)");
        $stmt->execute([$id]);
        $row = $stmt->fetch(PDO::FETCH_ASSOC);
        return $row ? UsuarioMapper::mapRowToDTO($row, $omitPassword)
: null;
    } catch (PDOException $e) {
        error_log("Error al leer usuario: " . $e->getMessage());
        return null;
    }
}

```

```

        }

    /**
     * Actualizar un usuario existente
     *
     * @param UsuarioDTO $usuario Objeto con los datos actualizados
     * @return bool True si se actualizó correctamente
     */
    public function update($usuario)
    {
        try {
            $stmt = $this->conn->prepare("CALL UsuarioUpdate(?, ?, ?, ?)");
            return $stmt->execute([
                $usuario->id,
                $usuario->usuario,
                $usuario->contrasena,
                $usuario->rol
            ]);
        } catch (PDOException $e) {
            error_log("Error al actualizar usuario: " . $e-
>getMessage());
            return false;
        }
    }
}

```

### |   |— UsuarioDTO.php

```

<?php
// Archivo: model.usuario/UsuarioDTO.php

/**
 * Clase UsuarioDTO
 *
 * Objeto de Transferencia de Datos (DTO) para la entidad Usuario.
 * Representa un registro de la tabla `usuario` y se utiliza
 * para transportar información entre la base de datos y la aplicación.
 */
class UsuarioDTO
{
    // Identificador único del usuario (PK)
    public $id;

    // Nombre de usuario (login)
    public $usuario;

```

```

    // Contraseña (almacenada en hash en la BD)
    public $contrasena;

    // Rol del usuario dentro del sistema (ej. ADMIN, USUARIO)
    public $rol;
}

```

| └─ UsuarioMapper.php

```

<?php
// Archivo: model.usuario/UsuarioMapper.php
require_once 'UsuarioDTO.php';

/**
 * Clase UsuarioMapper
 *
 * Se encarga de transformar filas de la base de datos en objetos
UsuarioDTO.
 * Permite desacoplar la estructura de la tabla de usuarios de la lógica
de la aplicación.
 */
class UsuarioMapper
{
    /**
     * Convierte una fila asociativa en un objeto UsuarioDTO.
     *
     * @param array $row Fila obtenida de la base de datos.
     * @param bool $omitPassword Indica si debe omitirse el campo
contraseña (true por defecto).
     * @return UsuarioDTO Objeto DTO con los datos mapeados.
     */
    public static function mapRowToDTO($row, $omitPassword = true)
    {
        $dto = new UsuarioDTO();
        $dto->id      = $row['Id'] ?? null;
        $dto->usuario = $row['Usuario'] ?? null;

        // Solo asigna la contraseña si no se omite (ej. login)
        if (!$omitPassword) {
            $dto->contraseña = $row['Contrasena'] ?? null;
        }

        $dto->rol = $row['Rol'] ?? null;
        return $dto;
    }
}

```

```

    |
    └── public/
        └── css/
            └── Analisis.css

/*
Analisis.css
-----
- Propósito: estilos para la vista de análisis (tarjetas, encabezado, botones, tablas y contenedores de gráfico/tabla).
- Uso: aplicar las clases descritas en los selectores para obtener el diseño moderno y responsive.
*/

/* VARIABLES GLOBALES (usar estas variables para mantener consistencia de colores y sombras) */
:root {
    /* Colores y sombras principales usados en el diseño */
    --amarillo: #f9c41f;
    --negro: #121212;
    --blanco: #ffffff;
    --gris: #838886;
    --gris-claro: #e0e0e0;
    --gris-muy-claro: #f5f5f5;
    --gris-oscuro: #666666;
    --gris-fondo: #f7f7f7;
    --naranja: #ff8c00;
    --fondo-blanco: rgba(255, 255, 255, 0.95);
    --sombra: rgba(0, 0, 0, 0.1);
    --sombra-hover: rgba(0, 0, 0, 0.15);
    --sombra-moderna: rgba(0, 0, 0, 0.06);
    --sombra-elevada: 0 8px 32px rgba(0, 0, 0, 0.12);
    --header-h: 72px;
    --card-max-w: 1200px;
}

.card {
    /* .card: contenedor base para paneles/ tarjetas de análisis
       - Uso: envuelve encabezado y contenido; aporta sombra, borde y fondo semi-transparente.
       - Recomendación: reutilizar para nuevos paneles para mantener consistencia visual. */
    border-radius: 20px;
    overflow: hidden;
    box-shadow: var(--sombra-elevada);
    margin-top: 2rem;
    border: 1px solid var(--gris-muy-claro);
    background: var(--fondo-blanco);
    backdrop-filter: blur(10px);
}

```

```

        transition: all 0.3s ease;
        position: relative;
    }

.card::before {
    /* Línea decorativa superior: degradado horizontal */
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    height: 4px;
    background: linear-gradient(90deg, var(--amarillo), var(--naranja));
    border-radius: 20px 20px 0 0;
}

.card:hover {
    transform: translateY(-4px);
    box-shadow: 0 16px 48px rgba(0, 0, 0, 0.15);
}

.card-header {
    /* Encabezado de la tarjeta
     - Degradado oscuro con texto amarillo.
     - Contiene títulos y controles del header. */
    background: linear-gradient(135deg, var(--negro), #2a2a2a)
!important;
    color: var(--amarillo) !important;
    font-weight: 700;
    font-size: 1.2rem;
    padding: 2rem 1.5rem;
    border: none;
    position: relative;
    overflow: hidden;
}

.card-header::before {
    /* Superposición sutil para dar textura al header sin afectar
legibilidad */
    content: '';
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    background: linear-gradient(45deg,
        transparent 30%,
        rgba(249, 196, 31, 0.05) 50%,

```

```

        transparent 70%);
pointer-events: none;
}

/* Título principal - Modernizado */
.header-title {
    /* Título principal del panel
     - Incluye ícono opcional en <i>
     - Color destacado y sombra para visibilidad */
    font-size: 1.5rem;
    font-weight: 700;
    margin-bottom: 1.5rem;
    display: flex;
    align-items: center;
    gap: 1rem;
    color: var(--amarillo);
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);
    letter-spacing: 0.5px;
    position: relative;
    z-index: 1;
}

.header-title i {
    font-size: 1.6rem;
    filter: drop-shadow(0 2px 4px rgba(0, 0, 0, 0.2));
}

/* Contenedor principal de controles - Modernizado */
.header-controls {
    /* Contenedor flex que organiza botones, buscador y acciones del
header */
    display: flex;
    flex-wrap: wrap;
    align-items: center;
    gap: 1.5rem;
    justify-content: space-between;
    position: relative;
    z-index: 1;
}

/* Grupo de botones de análisis - Modernizado */
.botones-analisis {
    /* Agrupa botones de filtros/acciones del análisis. Soporta wrap en
pantallas pequeñas. */
    display: flex;
    flex-wrap: wrap;
    gap: 0.75rem;
    flex: 1;
}

```

```

        min-width: 0;
    }

/* Acciones del header - Modernizado */
.acciones-header {
    /* Pequeño contenedor para acciones secundarias (ej: actualizar,
    exportar) */
    display: flex;
    align-items: center;
    flex-shrink: 0;
    gap: 1rem;
}

/* Contenedor del buscador - Modernizado */
.buscador-container {
    /* Contenedor del buscador (input y botón) */
    display: flex;
    align-items: center;
    flex-shrink: 0;
    position: relative;
    z-index: 1;
}

/* Estilos de los botones de análisis - Modernizados */
.btn-analisis {
    /* Botones principales de filtro/acción en el header
       - Uso: aplicar clase al elemento <button> o <a>
       - Efecto: brillo animado con pseudo-elemento y estados
    :hover/:active */
    border-radius: 12px;
    border: 2px solid var(--amarillo);
    color: var(--amarillo);
    background: transparent;
    font-weight: 600;
    padding: 0.75rem 1.25rem;
    transition: all 0.3s ease;
    font-size: 0.9rem;
    display: flex;
    align-items: center;
    gap: 0.5rem;
    white-space: nowrap;
    position: relative;
    overflow: hidden;
    backdrop-filter: blur(8px);
    box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
}
.btn-analisis::before {

```

```

/* Pseudo-elemento para efecto de brillo al pasar el ratón */
content: "";
position: absolute;
top: 0;
left: -100%;
width: 100%;
height: 100%;
background: linear-gradient(90deg, transparent, rgba(249, 196, 31, 0.2), transparent);
transition: left 0.5s ease;
}

.btn-analisis:hover::before {
    left: 100%;
}

.btn-analisis i {
    font-size: 0.9rem;
    transition: transform 0.3s ease;
}

.btn-analisis:hover i {
    transform: scale(1.1);
}

.btn-analisis {
    text-align: center;
    display: flex;
    align-items: center;
    justify-content: center;
}

.btn-analisis.active,
.btn-analisis:hover,
.btn-analisis:focus-visible {
    /* Estado activo/hover: degradado de marca y elevación */
    background: linear-gradient(135deg, var(--amarillo), var(--naranja));
    color: var(--negro);
    border-color: var(--amarillo);
    transform: translateY(-3px);
    box-shadow: 0 8px 20px rgba(249, 196, 31, 0.4);
}

/* Botón de actualizar - Modernizado */
.btn-actualizar {
    /* Botón alternativo con estilo claro (blanco) */
    border-radius: 12px;
    border: 2px solid var(--blanco);
}

```

```

color: var(--blanco);
background: transparent;
font-weight: 600;
padding: 0.75rem 1.25rem;
transition: all 0.3s ease;
font-size: 0.9rem;
display: flex;
align-items: center;
gap: 0.5rem;
white-space: nowrap;
position: relative;
overflow: hidden;
backdrop-filter: blur(8px);
box-shadow: 0 2px 8px rgba(255, 255, 255, 0.1);
}

.btn-actualizar::before {
    /* Efecto de brillo para botón actualizar */
    content: "";
    position: absolute;
    top: 0;
    left: -100%;
    width: 100%;
    height: 100%;
    background: linear-gradient(90deg, transparent, rgba(255, 255, 255, 0.2), transparent);
    transition: left 0.5s ease;
}

.btn-actualizar:hover::before {
    left: 100%;
}

.btn-actualizar:hover,
.btn-actualizar:focus-visible {
    background: linear-gradient(135deg, var(--blanco), #f0f0f0);
    color: var(--negro);
    border-color: var(--blanco);
    transform: translateY(-3px);
    box-shadow: 0 8px 20px rgba(255, 255, 255, 0.4);
}

/* Buscador - Modernizado */
.buscador-analisis {
    /* Componente buscador: input con icono y estilo de tarjeta */
    min-width: 320px;
    max-width: 380px;
    border-radius: 12px;
}

```

```

        overflow: hidden;
        box-shadow: var(--sombra-elevada);
        border: 1px solid var(--gris-muy-claro);
        transition: all 0.3s ease;
    }

    .buscador-analisis:hover {
        transform: translateY(-2px);
        box-shadow: 0 12px 32px rgba(0, 0, 0, 0.15);
    }

    .input-group-text {
        /* Estilo del icono/label dentro del input-group */
        background: linear-gradient(135deg, var(--amarillo), var(--naranja)) !important;
        color: var(--negro) !important;
        border: none;
        font-weight: 700;
        padding: 1rem 1.25rem;
        text-shadow: 0 1px 2px rgba(0, 0, 0, 0.1);
    }

    .buscador-analisis .form-control {
        /* Input principal del buscador: fondo blanco y foco con sutil escala */
        background: var(--blanco);
        color: var(--negro);
        font-weight: 500;
        border: none;
        padding: 1rem 1.25rem;
        font-size: 0.95rem;
        transition: all 0.3s ease;
    }

    .buscador-analisis .form-control:focus {
        outline: none;
        box-shadow: none;
        background: var(--blanco);
        transform: scale(1.02);
    }

    .buscador-analisis .form-control::placeholder {
        color: var(--gris-oscuro);
        font-weight: 400;
    }

    /* Estilos de tabla - Modernizados */
    .btn-outline-warning {

```

```

/* Botón estilo outline para acciones dentro de tablas */
border-radius: 12px;
border-width: 2px;
font-weight: 600;
color: var(--amarillo);
border-color: var(--amarillo);
background: transparent;
transition: all 0.3s ease;
padding: 0.5rem 1rem;
box-shadow: 0 2px 8px rgba(249, 196, 31, 0.1);
}

.btn-outline-warning:hover,
.btn-outline-warning.active {
    background: linear-gradient(135deg, var(--amarillo), var(--naranja));
    color: var(--negro);
    border-color: var(--amarillo);
    transform: translateY(-2px);
    box-shadow: 0 6px 16px rgba(249, 196, 31, 0.3);
}

.table {
    /* Tablas con bordes redondeados y encabezado sticky */
    border-radius: 16px;
    overflow: hidden;
    margin-bottom: 0;
    box-shadow: var(--sombra-moderna);
    border: 1px solid var(--gris-muy-claro);
}

.table thead th {
    /* Encabezados con degradado de marca y posición sticky para scroll */
    background: linear-gradient(135deg, var(--amarillo), var(--naranja)) !important;
    color: var(--negro) !important;
    font-weight: 700;
    border: none;
    padding: 1rem 0.75rem;
    text-shadow: 0 1px 2px rgba(0, 0, 0, 0.1);
    position: sticky;
    top: 0;
    z-index: 10;
}

.table-striped tbody tr:nth-of-type(odd) {
    background: linear-gradient(135deg, #fff9e6, #ffffbf0);
}

```

```

.table-hover tbody tr:hover {
    background: linear-gradient(135deg, #fef3c7, #fde68a) !important;
    transform: translateY(-1px);
    box-shadow: 0 4px 12px rgba(249, 196, 31, 0.1);
    transition: all 0.3s ease;
}

.table tbody tr {
    border-bottom: 1px solid var(--gris-muy-claro);
    transition: all 0.3s ease;
}

.table td {
    padding: 0.75rem;
    vertical-align: middle;
}

#analisisGraficoCont {
    /* Contenedor del gráfico: fondo claro, padding amplio y borde
decorativo superior */
    background: linear-gradient(135deg, #fffbeb, #fefce8);
    border-radius: 16px;
    margin-top: 2rem;
    padding: 2rem;
    border: 1px solid var(--gris-muy-claro);
    box-shadow: var(--sombra-moderna);
    position: relative;
    overflow: hidden;
}

#analisisGraficoCont::before {
    /* Línea decorativa superior del contenedor del gráfico */
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    height: 3px;
    background: linear-gradient(90deg, var(--amarillo), var(--naranja));
    border-radius: 16px 16px 0 0;
}

#analisisTablaCont {
    /* Contenedor de la tabla con scroll vertical y scrollbar
personalizado */
    max-height: 400px;
    overflow-y: auto;
}

```

```
background: var(--blanco);
border-radius: 16px;
padding-bottom: 1rem;
border: 1px solid var(--gris-muy-claro);
box-shadow: var(--sombra-moderna);
}

#analisisTablaCont::-webkit-scrollbar {
    width: 8px;
}

#analisisTablaCont::-webkit-scrollbar-track {
    background: var(--gris-muy-claro);
}

#analisisTablaCont::-webkit-scrollbar-thumb {
    background: var(--amarillo);
    border-radius: 4px;
}

#analisisTablaCont::-webkit-scrollbar-thumb:hover {
    background: var(--naranja);
}

.header-line {
    display: flex;
    align-items: center;
    gap: 1.5rem;
    min-width: 0;
}

.header-title {
    font-size: 1.5rem;
    font-weight: 700;
    color: var(--amarillo);
    display: flex;
    align-items: center;
    gap: 1rem;
    white-space: nowrap;
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);
    letter-spacing: 0.5px;
}

.divider-header {
    /* Línea/divisor principal del header con degradado */
    height: 4px;
    background: linear-gradient(90deg, var(--amarillo) 0%, var(--naranja) 70%, transparent 100%);
}
```

```

border-radius: 2px;
flex-grow: 1;
min-width: 60px;
margin-left: 1.5rem;
opacity: 0.9;
box-shadow: 0 2px 4px rgba(249, 196, 31, 0.2);
}

/* ====== RESPONSIVE MOBILE - Modernizado ====== */

/* Tablets y móviles grandes */
@media (max-width: 991.98px) {
    /* Ajustes para tablets y móviles grandes: reducir márgenes y tamaños */
    .card {
        margin: 1rem;
        border-radius: 16px;
    }

    .card-header {
        padding: 1.5rem 1rem;
    }

    .header-title {
        font-size: 1.3rem;
        margin-bottom: 1.25rem;
    }

    .header-title i {
        font-size: 1.4rem;
    }

    .buscador-analisis {
        min-width: 280px;
        max-width: 320px;
    }

    .header-controls {
        gap: 1.25rem;
    }

    .botones-analisis {
        gap: 0.5rem;
    }
}

/* Móviles medianos */
@media (max-width: 768px) {

```

```
/* Ajustes para móviles medianos: botones y buscador más compactos */
.card {
    margin: 0.75rem;
    border-radius: 12px;
}

.card-header {
    padding: 1rem 0.75rem;
}

.header-title {
    font-size: 1.2rem;
    margin-bottom: 1rem;
}

.header-title i {
    font-size: 1.3rem;
}

/* Botones más pequeños en móvil */
.btn-analisis {
    font-size: 0.85rem;
    padding: 0.5rem 1rem;
    border-radius: 10px;
}

.btn-analisis i {
    font-size: 0.85rem;
}

.btn-actualizar {
    font-size: 0.85rem;
    padding: 0.5rem 1rem;
    border-radius: 10px;
}

/* Buscador más pequeño */
.buscador-analisis {
    min-width: 240px;
    max-width: 280px;
}

.input-group-text {
    padding: 0.75rem 1rem;
}

.buscador-analisis .form-control {
    padding: 0.75rem 1rem;
```

```
        font-size: 0.9rem;
    }

/* Margenes más pequeños */
.botones-analisis {
    gap: 0.5rem;
}

.header-controls {
    gap: 1rem;
}

/* Tabla responsive en móvil */
.table {
    font-size: 0.9rem;
}

.table th,
.table td {
    padding: 0.75rem 0.5rem;
}

/* Gráfico más pequeño */
#analisisGraficoCont {
    margin-top: 1.5rem;
    padding: 1.5rem;
    min-height: 280px;
}

#analisisTablaCont {
    max-height: 320px;
    padding-bottom: 0.75rem;
}
}

/* Móviles pequeños */
@media (max-width: 576px) {
    /* Ajustes para móviles pequeños: ocultar columnas menos relevantes y
    compactar todo */
    .card {
        margin: 0.5rem;
        border-radius: 10px;
    }

    .card-header {
        padding: 0.75rem 0.5rem;
    }
}
```

```
.header-title {  
    font-size: 1.1rem;  
    margin-bottom: 0.75rem;  
    gap: 0.75rem;  
}  
  
.header-title i {  
    font-size: 1.2rem;  
}  
  
/* Título y línea divisoria más compactos */  
.divider-header {  
    height: 3px;  
    margin-left: 0.75rem;  
    min-width: 30px;  
}  
  
/* Botones aún más pequeños */  
.btn-analisis {  
    font-size: 0.8rem;  
    padding: 0.4rem 0.8rem;  
    border-radius: 8px;  
    gap: 0.4rem;  
}  
  
.btn-analisis i {  
    font-size: 0.8rem;  
}  
  
.btn-actualizar {  
    font-size: 0.8rem;  
    padding: 0.4rem 0.8rem;  
    border-radius: 8px;  
    gap: 0.4rem;  
}  
  
/* Buscador en móvil pequeño */  
.buscador-analisis {  
    min-width: 200px;  
    max-width: 240px;  
    border-radius: 10px;  
}  
  
.input-group-text {  
    padding: 0.6rem 0.8rem;  
    font-size: 0.85rem;  
}
```

```
.buscador-analisis .form-control {
    padding: 0.6rem 0.8rem;
    font-size: 0.85rem;
}

/* Ajustar espaciado en filas de botones */
.botones-analisis {
    gap: 0.4rem;
    margin-bottom: 0.75rem !important;
}

.header-controls {
    gap: 0.75rem;
}

/* Tabla muy compacta */
.table {
    font-size: 0.8rem;
}

.table th,
.table td {
    padding: 0.5rem 0.3rem;
    vertical-align: middle;
}

/* Ocultar columnas menos importantes en móvil pequeño */
.table th:nth-child(3),
.table td:nth-child(3),
.table th:nth-child(4),
.table td:nth-child(4) {
    display: none;
}

/* Gráfico mínimo */
#analisisGraficoCont {
    margin-top: 1rem;
    padding: 1rem;
    min-height: 220px;
}

#analisisTablaCont {
    max-height: 280px;
}

/* Botones de colapso más pequeños */
.btn-sm {
    font-size: 0.75rem;
```

```

        padding: 0.3rem 0.6rem;
        border-radius: 6px;
    }

    /* Alertas más compactas */
    .alert {
        font-size: 0.85rem;
        padding: 0.75rem;
        margin-top: 1rem !important;
        border-radius: 8px;
    }
}

```

### | | └ Bitacora.css

```

/*
Bitacora.css
-----
- Propósito: estilos específicos para la página de bitácora (fondo,
contenedor principal, tablas y botones).
- Uso: mantener las clases tal cual en el HTML para que los estilos y
comportamientos responsivos se apliquen correctamente.
*/

body {
    background: #838886 !important;
    font-family: "Segoe UI", sans-serif;
    margin: 0;
    color: #333;
    position: relative;
    min-height: 100vh;
}

/* Fondo decorativo - igual que las otras secciones */
body::before,
body::after {
    content: "";
    position: fixed;
    z-index: -1;
    pointer-events: none;
}

body::before {
    top: 0;
    left: 0;
    width: 40%;
    height: 70%;
    background: #f9c41f;
}

```

```
body::after {
    bottom: 0;
    right: 0;
    width: 60%;
    height: 50%;
    background: #838886;
}

/* Contenedor centrado con efectos mejorados */
.contenedor-bitacora {
    margin: 0 auto;
    padding: 40px;
    background: rgba(255, 255, 255, 0.95);
    border-radius: 20px;
    box-shadow: 0 8px 32px rgba(0, 0, 0, 0.12);
    backdrop-filter: blur(10px);
    position: relative;
    max-width: 1200px;
    overflow: hidden;
}

/* Línea superior decorativa */
.contenedor-bitacora::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 4px;
    background: linear-gradient(90deg, #f9c41f 0%, #ffd700 100%);
    z-index: 1;
}

/* Contenedor de tabla con scroll personalizado */
.table-responsive {
    width: 100%;
    overflow-x: auto;
    border-radius: 16px;
    -webkit-overflow-scrolling: touch;
}

/* Scroll personalizado */
.table-responsive::-webkit-scrollbar {
    width: 8px;
    height: 8px;
}
```

```
.table-responsive::-webkit-scrollbar-track {
    background: rgba(255, 255, 255, 0.1);
    border-radius: 8px;
}

.table-responsive::-webkit-scrollbar-thumb {
    background: linear-gradient(135deg, #f9c41f, #ffd700);
    border-radius: 8px;
    box-shadow: 0 2px 8px rgba(249, 196, 31, 0.3);
}

.table-responsive::-webkit-scrollbar-thumb:hover {
    background: linear-gradient(135deg, #e6b619, #f9c41f);
}

/* Alerta con efectos mejorados */
.alert {
    background: linear-gradient(135deg, #2c2c2c 0%, #1a1a1a 100%);
    color: #f9c41f;
    padding: 1rem 1.5rem;
    border-left: 6px solid #f9c41f;
    border-radius: 12px;
    margin-bottom: 24px;
    box-shadow: 0 4px 16px rgba(0, 0, 0, 0.1);
    transition: all 0.3s ease;
}

.alert:hover {
    transform: translateY(-2px);
    box-shadow: 0 8px 24px rgba(249, 196, 31, 0.2);
}

/* Botones con efectos mejorados */
.btn {
    padding: 14px 28px;
    font-size: 1rem;
    font-weight: 600;
    border: none;
    border-radius: 12px;
    margin: 8px 12px;
    cursor: pointer;
    transition: all 0.3s ease;
    text-transform: uppercase;
    letter-spacing: 0.5px;
    position: relative;
    overflow: hidden;
}
```

```

.btn:hover {
    transform: translateY(-2px);
    box-shadow: 0 8px 24px rgba(0, 0, 0, 0.2);
}

.btn-pdf {
    background: linear-gradient(135deg, #e74c3c, #c0392b);
    color: white;
    box-shadow: 0 4px 16px rgba(231, 76, 60, 0.3);
}

.btn-pdf:hover {
    background: linear-gradient(135deg, #c0392b, #a93226);
    box-shadow: 0 8px 24px rgba(231, 76, 60, 0.4);
}

.btn-excel {
    background: linear-gradient(135deg, #2ecc71, #27ae60);
    color: white;
    box-shadow: 0 4px 16px rgba(46, 204, 113, 0.3);
}

.btn-excel:hover {
    background: linear-gradient(135deg, #27ae60, #229954);
    box-shadow: 0 8px 24px rgba(46, 204, 113, 0.4);
}

/* Tabla con efectos de hover como las otras secciones */
table {
    width: 100%;
    border-collapse: separate;
    border-spacing: 0;
    background-color: #ffffff;
    margin-top: 1rem;
    border-radius: 16px;
    overflow: hidden;
    box-shadow: 0 8px 32px rgba(0, 0, 0, 0.12);
    table-layout: fixed;
}

th {
    background: linear-gradient(135deg, #f9c41f 0%, #ffd700 100%);
    color: #000;
    font-weight: bold;
    padding: 16px 12px;
    text-align: center;
    text-transform: uppercase;
    font-size: 0.9rem;
}

```

```

letter-spacing: 0.5px;
position: sticky;
top: 0;
z-index: 10;
}

td {
  padding: 14px 12px;
  text-align: center;
  border-bottom: 1px solid rgba(0, 0, 0, 0.1);
  background-color: rgba(255, 255, 255, 0.8);
  color: #333;
  transition: all 0.3s ease;
  vertical-align: middle;
}

/* Efectos de hover para las filas - igual que otras secciones */
tbody tr {
  transition: all 0.3s ease;
  border-bottom: 1px solid rgba(0, 0, 0, 0.1);
}

tbody tr:hover {
  background: linear-gradient(
    135deg,
    rgba(249, 196, 31, 0.15),
    rgba(255, 255, 255, 0.95)
  );
  transform: translateY(-2px);
  box-shadow: 0 6px 20px rgba(249, 196, 31, 0.2);
}

tbody tr:hover td {
  background: transparent !important;
  color: #000 !important;
}

tbody tr:nth-child(even) td {
  background-color: rgba(248, 249, 250, 0.6);
}

tbody tr:nth-child(even):hover td {
  background: transparent !important;
  color: #000 !important;
}

td:first-child {
  text-align: left;
}

```

```
    font-weight: 500;
    padding-left: 20px;
}

@media (max-width: 768px) {
    .contenedor-bitacora {
        margin: 12px;
        padding: 20px;
    }

    body::before {
        width: 60%;
        height: 50%;
    }

    body::after {
        width: 55%;
        height: 40%;
    }
}

.btn {
    width: 100%;
    margin: 8px 0;
    font-size: 0.9rem;
}

table {
    font-size: 0.85rem;
}

th,
td {
    padding: 8px 6px;
}
}

@media (max-width: 480px) {
    body::before,
    body::after {
        display: none;
    }

    .contenedor-bitacora {
        margin: 4px;
        padding: 12px;
        border-radius: 12px;
    }
}
```

```

/* === OVERRIDE ESPECÍFICO PARA BITÁCORA === */
/* Forzar fondo blanco en la página de bitácora */
body[class*="bitacora"],
body[data-page="bitacora"],
.bitacora-page,
.bitacora body,
body {
    background: #ffffff !important;
    background-color: #ffffff !important;
}

html {
    background: #ffffff !important;
    background-color: #ffffff !important;
}

/* SIDEBAR MODERNO */
.sidebar {
    width: var(--sidebar-width);
    background: linear-gradient(135deg, var(--negro) 0%, #1a1a1a 100%);
    color: var(--blanco);
    padding: 24px 20px;
    position: fixed;
    top: 0;
    bottom: 0;
    left: 0;
    overflow-y: auto;
    box-shadow: var(--sombra-elevada);
    z-index: 1000;
    transition: all 0.3s ease;
}

/* Hacer scroll invisible en el sidebar */
.sidebar {
    scrollbar-width: none; /* Firefox */
    -ms-overflow-style: none; /* IE y Edge */
}

.sidebar::-webkit-scrollbar {
    display: none; /* Chrome, Safari y Opera */
}

/* HEADER DEL SIDEBAR MODERNIZADO */
.sidebar-header {
    display: flex;
    align-items: center;
    gap: 12px;
    margin-bottom: 2rem;
}

```

```

padding: 16px 12px;
border-radius: 12px;
background: rgba(249, 196, 31, 0.1);
border: 1px solid rgba(249, 196, 31, 0.2);
position: relative;
overflow: hidden;
}

.sidebar-header::before {
content: "";
position: absolute;
top: 0;
left: 0;
width: 100%;
height: 2px;
background: linear-gradient(90deg, var(--amarillo) 0%, #ffd700 100%);
}
}

td:first-child {
text-align: left;
padding-left: 20px;
white-space: normal; /*  Permite saltos de línea */
word-wrap: break-word; /*  Parte palabras largas si es necesario */
*/
overflow-wrap: break-word; /*  Compatibilidad amplia */
}
}

@media screen and (max-width: 480px) {
td:first-child {
white-space: normal !important;
word-break: break-word;
text-align: left;
padding-left: 12px;
}
}
}

```

### | | └─ Cliente.css

```

/*
Cliente.css
-----
- Propósito: estilos para la sección de clientes (lista, tarjetas,
tablas, botones y buscador).
- Uso: mantener las clases descritas en el HTML para que el diseño y el
comportamiento responsive se apliquen.
- Variables: se usan variables CSS en :root (colores y sombras) para
mantener coherencia visual.
*/

```

```

#historialReasignaciones {
    max-height: 300px;
    overflow-y: auto;
    padding-right: 10px;
    scrollbar-width: none; /* Firefox */
    -ms-overflow-style: none; /* IE y Edge */
}
#historialReasignaciones::-webkit-scrollbar {
    display: none; /* Chrome, Safari, Opera */
}

/* Scroll invisible pero funcional en todas las tablas con scroll */
.table-responsive {
    scrollbar-width: none; /* Firefox */
    -ms-overflow-style: none; /* IE y Edge */
}
.table-responsive::-webkit-scrollbar {
    display: none; /* Chrome, Safari, Opera */
}
/* Cliente.css - versión moderna con fondo decorativo y sombras avanzadas */
*/
:root {
    /* Variables globales de color y sombra: cambiar aquí para ajustar la paleta */
    --amarillo: #f9c41f;
    --negro: #121212;
    --blanco: #ffffff;
    --gris: #838886;
    --gris-claro: #e0e0e0;
    --gris-muy-claro: #f5f5f5;
    --gris-oscuro: #666666;
    --gris-fondo: #f7f7f7;
    --naranja: #ff8c00;
    --fondo-blanco: rgba(255, 255, 255, 0.95);
    --sombra: rgba(0, 0, 0, 0.1);
    --sombra-hover: rgba(0, 0, 0, 0.15);
    --sombra-moderna: rgba(0, 0, 0, 0.06);
    --sombra-elevada: 0 8px 32px rgba(0, 0, 0, 0.12);
}
* {
    box-sizing: border-box;
}

/* ===== Layout base modernizado ===== */
body {

```

```

/* Contenedor base del layout:
   - 'display: flex' usado para permitir sidebar fijo a la izquierda.
   - 'background' usa variable global; si se requiere tema claro,
cambiar en :root.
*/
margin: 0;
font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
display: flex;
min-height: 100vh;
background: var(--gris) !important;
overflow-x: hidden;
}

body::-webkit-scrollbar {
  display: none;
}

/* Fondo decorativo moderno */
body::before,
body::after {
  content: "";
  position: fixed;
  z-index: -1;
  pointer-events: none;
}

body::before {
  top: 0;
  left: 0;
  width: 40%;
  height: 70%;
  background: var(--amarillo);
}

body::after {
  bottom: 0;
  right: 0;
  width: 60%;
  height: 50%;
  background: var(--gris);
}

/* ===== Contenido principal modernizado ===== */
.content {
  /* Área principal donde se renderiza la tabla y las tarjetas de
cliente. */
  margin-left: 220px;
  padding: 24px;
}

```

```

    flex: 1;
}

.container {
  max-width: 102%;
  transform: translateX(1%);
}

/* ===== Botones modernos ===== */
.btn-primary {
  /* Botón primario (ej: crear/añadir cliente)
   - Usar clase .btn-primary en botones principales.
   - Tiene estados :hover y :focus definidos para accesibilidad visual.
*/
  background: linear-gradient(135deg, var(--amarillo) 0%, #ffd700 100%);
  color: var(--negro);
  font-weight: 600;
  border: none;
  border-radius: 8px;
  padding: 12px 24px;
  font-size: 0.95rem;
  box-shadow: 0 4px 12px rgba(249, 196, 31, 0.3);
  transition: all 0.3s ease;
  cursor: pointer;
}

.btn-primary:hover {
  background: linear-gradient(135deg, #e6b01f 0%, #ffcc00 100%);
  transform: translateY(-2px);
  box-shadow: 0 6px 20px rgba(249, 196, 31, 0.4);
}

.btn-primary:focus {
  outline: none;
  box-shadow: 0 0 0 3px rgba(249, 196, 31, 0.2);
}

.btn-danger {
  /* Botón de peligro/ eliminación */
  background: linear-gradient(135deg, #e74c3c 0%, #c0392b 100%);
  color: var(--blanco);
  font-weight: 600;
  border: none;
  border-radius: 8px;
  padding: 12px 24px;
  font-size: 0.95rem;
  box-shadow: 0 4px 12px rgba(231, 76, 60, 0.3);
  transition: all 0.3s ease;
}

```

```

        cursor: pointer;
    }

.btn-danger:hover {
    background: linear-gradient(135deg, #c0392b 0%, #a93226 100%);
    transform: translateY(-2px);
    box-shadow: 0 6px 20px rgba(231, 76, 60, 0.4);
}

/* ===== Cards modernas ===== */
.card {
    /* Tarjetas reutilizables para secciones de contenido
       - Reutilizar .card para paneles de detalle o formularios. */
    border: none;
    border-radius: 16px;
    background: rgba(255, 255, 255, 0.95);
    overflow: hidden;
    box-shadow: var(--sombra-elevada);
    margin-bottom: 24px;
    transition: all 0.3s ease;
    position: relative;
}

.card::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 4px;
    background: linear-gradient(90deg, var(--amarillo) 0%, #ffd700 100%);
}

.card:hover {
    transform: translateY(-4px);
    box-shadow: 0 12px 40px rgba(0, 0, 0, 0.15);
}

.card.shadow.mb-4 {
    box-shadow: var(--sombra-elevada);
    border-left: none;
}

.card-header {
    background: linear-gradient(135deg, var(--negro) 0%, #1a1a1a 100%) !important;
    color: var(--blanco) !important;
    font-weight: 600;
}

```

```

letter-spacing: 0.5px;
padding: 20px 24px;
border: none;
}

/* Lista de clientes con scroll */
#clienteLista {
    /* Contenedor reciclable para listas con desplazamiento interno */
    max-height: 300px;
    overflow-y: auto;
    padding-right: 10px;
    scrollbar-width: none;
}

#clienteLista::-webkit-scrollbar {
    display: none;
}

/* Tabla con scroll horizontal - Modernizada */
.table-container {
    /* Contenedor de tabla responsive
       - Mantiene fondo, bordes y sombra.
       - Utilizar para tablas grandes con scroll-x. */
    width: 100%;
    overflow-x: auto;
    -webkit-overflow-scrolling: touch;
    margin-bottom: 1.5rem;
    border-radius: 16px;
    background: var(--fondo-blanco);
    box-shadow: var(--sombra-elevada);
    border: 1px solid var(--gris-muy-claro);
    backdrop-filter: blur(8px);
    position: relative;
}

.table-container::before {
    /* Línea decorativa superior para dar consistencia visual con otras
       tarjetas */
    content: '';
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    height: 4px;
    background: linear-gradient(90deg, var(--amarillo), var(--naranja));
    border-radius: 16px 16px 0 0;
}

```

```
.celda-correo {
    /* Asegura que direcciones de correo se rompan correctamente en varias
líneas */
    white-space: normal;
    word-break: break-word;
    overflow-wrap: break-word;
}

.table-container table {
    /* Tabla interna: usar min-width para permitir scroll horizontal en
pantallas pequeñas */
    width: 100%;
    table-layout: fixed;
    min-width: 700px;
    white-space: nowrap;
    border-collapse: separate;
    border-spacing: 0;
    background: transparent;
}

.table-container th,
.table-container td {
    /* Celdas: truncamiento con ellipsis y padding consistente */
    overflow: hidden;
    text-overflow: ellipsis;
    max-width: 250px;
    padding: 16px 20px;
    border-bottom: 1px solid var(--gris-muy-claro);
    transition: all 0.3s ease;
}

.table-container th {
    background: linear-gradient(135deg, var(--gris-muy-claro), #f8f9fa);
    font-weight: 600;
    text-align: left;
    color: var(--negro);
    text-transform: uppercase;
    font-size: 0.85rem;
    letter-spacing: 0.5px;
    position: sticky;
    top: 0;
    z-index: 10;
}

.table-container th:first-child {
    border-radius: 0 0 0 16px;
}
```

```

.table-container th:last-child {
  border-radius: 0 0 16px 0;
}

.table-container tr:hover td {
  /* Hover en fila: resalta con degradado y ligero desplazamiento */
  background: linear-gradient(135deg, #fff9e6, #ffffbf0);
  transform: translateY(-1px);
  box-shadow: 0 4px 12px rgba(249, 196, 31, 0.1);
}

.table-container td:hover {
  overflow: visible;
  white-space: normal;
  word-break: break-all;
  z-index: 15;
  position: relative;
  background: var(--fondo-blanco);
  box-shadow: var(--sombra-elevada);
  border-radius: 8px;
}

/* Estilos para historial de reasignaciones - Modernizado */
.fila-cliente:hover {
  background: linear-gradient(135deg, rgba(249, 196, 31, 0.1), rgba(255, 255, 255, 0.9)) !important;
  transform: translateX(4px);
  transition: all 0.3s ease;
  box-shadow: 4px 0 12px rgba(249, 196, 31, 0.15);
}

.codigo-barra {
  /* Estilo para mostrar códigos o identificadores monoespaciados */
  font-family: 'JetBrains Mono', 'Courier New', monospace;
  background: linear-gradient(135deg, var(--gris-muy-claro), #f1f3f4);
  padding: 6px 12px;
  border-radius: 8px;
  font-size: 0.85em;
  color: var(--negro);
  border: 1px solid var(--gris-claro);
  display: inline-block;
  transition: all 0.3s ease;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.05);
}

.codigo-barra:hover {
  background: linear-gradient(135deg, var(--amarillo), #fff3cd);
  transform: translateY(-1px);
}

```

```
    box-shadow: 0 4px 8px rgba(249, 196, 31, 0.2);
}

.codigo-barra-small {
  font-family: 'JetBrains Mono', 'Courier New', monospace;
  background: linear-gradient(135deg, var(--gris-muy-claro), #f8f9fa);
  padding: 4px 8px;
  border-radius: 6px;
  font-size: 0.75em;
  color: var(--gris-oscuro);
  border: 1px solid var(--gris-claro);
  display: inline-block;
  transition: all 0.3s ease;
}

.codigo-barra-small:hover {
  background: linear-gradient(135deg, #e3f2fd, #f3e5f5);
  transform: scale(1.05);
}

.motivo-cell {
  max-width: 200px;
}

.motivo-texto {
  display: block;
  overflow: hidden;
  text-overflow: ellipsis;
  white-space: nowrap;
  max-width: 180px;
}

.detalle-reasignacion {
  /* Panel lateral/mini tarjeta que muestra detalle de reasignación */
  background: linear-gradient(135deg, #e3f2fd, #f8f9fa);
  border-left: 4px solid var(--amarillo);
  border-radius: 0 8px 8px 0;
  box-shadow: 0 2px 8px rgba(249, 196, 31, 0.1);
  transition: all 0.3s ease;
}

.detalle-reasignacion:hover {
  transform: translateX(2px);
  box-shadow: 0 4px 16px rgba(249, 196, 31, 0.15);
}

.alert-sm {
  /* Alertas pequeñas para mensajes de estado o información contextual */
```

```

padding: 12px 16px;
font-size: 0.875rem;
border-radius: 8px;
border: none;
background: linear-gradient(135deg, #d1ecf1, #bee5eb);
color: var(--negro);
box-shadow: 0 2px 8px rgba(0, 0, 0, 0.05);
}

.table th {
  border-bottom: 2px solid var(--gris-claro);
  font-weight: 600;
  font-size: 0.9em;
  background: linear-gradient(135deg, var(--gris-muy-claro), #f8f9fa);
  color: var(--negro);
}

.table td {
  vertical-align: middle;
  font-size: 0.9em;
  transition: all 0.3s ease;
}

.table tbody tr:hover td {
  background: linear-gradient(135deg, rgba(249, 196, 31, 0.05), rgba(255, 255, 255, 0.9));
}

.badge {
  /* Badges para estados cortos (ej: 'activo', 'nuevo') */
  font-size: 0.75em;
  padding: 6px 12px;
  border-radius: 12px;
  font-weight: 500;
  background: linear-gradient(135deg, var(--amarillo), #fff3cd);
  color: var(--negro);
  border: 1px solid rgba(249, 196, 31, 0.3);
  box-shadow: 0 2px 4px rgba(249, 196, 31, 0.1);
}

/* Responsive para móviles - Modernizado */
@media (max-width: 992px) {
  /* Ajustes en tablet: eliminar margen izquierdo para mostrar contenido
  a pantalla completa */
  .content {
    margin-left: 0;
    padding: 1rem;
    width: 100%;
  }
}

```

```
    }
}

@media (max-width: 768px) {
  body::before,
  body::after {
    display: none;
  }

  .content {
    margin-left: 0;
    padding: 1rem;
    width: 100%;
  }

  h2,
  .topbar h4 {
    text-align: center;
    font-size: 1.2rem;
    margin-bottom: 1rem;
    color: var(--negro);
  }

  .card {
    width: 95%;
    margin: 0 auto 1.5rem auto;
    border-radius: 16px;
    box-shadow: var(--sombra-elevada);
  }

  .card-header {
    font-size: 1rem;
    text-align: center;
    background: linear-gradient(135deg, var(--amarillo), #fff3cd);
    color: var(--negro);
    border-radius: 16px 16px 0 0;
    border: none;
  }

  .card.shadow.mb-4 {
    box-shadow: var(--sombra-elevada);
    padding: 1.5rem;
    border-radius: 16px;
    background: var(--fondo-blanco);
  }

  .container {
    max-width: 95%;
```

```
    margin: 0 auto;
    transform: none;
}

.table-container {
    width: 100%;
    overflow-x: auto;
    -webkit-overflow-scrolling: touch;
    margin-bottom: 1.5rem;
    border-radius: 12px;
    box-shadow: var(--sombra-elevada);
}

.table-container table {
    min-width: 600px;
}
}

@media (max-width: 576px) {
    .clientes-header {
        flex-direction: column;
        align-items: stretch;
        gap: 1rem;
        padding: 1rem !important;
    }

    .buscador-clientes {
        max-width: 100%;
        min-width: 0;
        border-radius: 12px;
        font-size: 0.9rem;
    }

    .titulo-clientes {
        font-size: 1rem;
        text-align: center;
    }

    .card {
        width: 100%;
        margin: 0 0 1rem 0;
        border-radius: 12px;
    }

    .table-container {
        border-radius: 8px;
        margin-bottom: 1rem;
    }
}
```

```

}

@media (max-width: 480px) {
    /* Ajustes para móviles pequeños: compactar paddings y tamaños */
    .content {
        padding: 0.5rem;
    }

    .card {
        margin: 0 0 0.75rem 0;
        border-radius: 8px;
    }

    .card-body {
        padding: 0.75rem;
    }

    .btn {
        font-size: 0.85rem;
        padding: 8px 12px;
    }

    .table-container th,
    .table-container td {
        padding: 8px 12px;
        font-size: 0.8rem;
    }
}

/* Header de clientes - Modernizado */
.clientes-header {
    /* Cabecera específica de la vista de clientes:
       - Degradado oscuro, borde inferior de marca y padding.
       - Usar para contener título, buscador y acciones principales. */
    background: linear-gradient(135deg, var(--negro), #1a1a1a)
!important;
    border-bottom: 4px solid var(--amarillo);
    padding: 1.5rem 2rem !important;
    box-shadow: var(--sombra-elevada);
    position: relative;
    overflow: hidden;
}

.clientes-header::before {
    content: '';
    position: absolute;
    top: 0;
    left: 0;
}

```

```

    right: 0;
    bottom: 0;
    background: linear-gradient(45deg,
        transparent 30%,
        rgba(249, 196, 31, 0.05) 50%,
        transparent 70%);
    pointer-events: none;
}

.titulo-clientes {
    /* Título destacado en el header de clientes */
    color: var(--amarillo);
    font-weight: 700;
    margin-bottom: 0;
    font-size: 1.2rem;
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);
    letter-spacing: 0.5px;
}

.buscador-clientes {
    /* Campo buscador principal de la vista de clientes
       - Tiene placeholder y estados :focus visuales. */
    max-width: 350px;
    min-width: 250px;
    font-weight: 500;
    border-radius: 24px;
    background: var(--blanco);
    color: var(--negro);
    box-shadow: 0 4px 16px rgba(0, 0, 0, 0.1);
    border: 2px solid var(--gris-claro);
    padding: 12px 20px;
    transition: all 0.3s ease;
    backdrop-filter: blur(8px);
    position: relative;
    z-index: 1;
}

.buscador-clientes::placeholder {
    color: var(--gris-oscuro);
    font-weight: 400;
    opacity: 1;
}

.buscador-clientes:focus {
    outline: none;
    box-shadow: 0 0 0 4px rgba(249, 196, 31, 0.3);
    border-color: var(--amarillo);
    transform: translateY(-2px);
}

```

```

        background: var(--blanco);
        color: var(--negro);
    }

/* Cuando el campo tiene contenido */
.buscador-clientes:not(:placeholder-shown) {
    background: var(--blanco);
    color: var(--negro);
    font-weight: 600;
}

/* Botón actualizar en el header - Modernizado */
.clientes-header .btn-outline-light {
    border: 2px solid var(--amarillo);
    color: var(--amarillo);
    background: transparent;
    font-weight: 600;
    border-radius: 12px;
    padding: 10px 20px;
    transition: all 0.3s ease;
    position: relative;
    z-index: 1;
    backdrop-filter: blur(8px);
}

.clientes-header .btn-outline-light:hover {
    background: var(--amarillo);
    color: var(--negro);
    transform: translateY(-2px);
    box-shadow: 0 8px 20px rgba(249, 196, 31, 0.3);
}

@media (max-width: 576px) {
    .clientes-header {
        flex-direction: column;
        align-items: stretch !important;
        gap: 1rem !important;
        padding: 1rem !important;
    }
}

.titulo-clientes {
    text-align: center;
    margin-bottom: 0.5rem;
}

.buscador-clientes {
    max-width: 100%;
    min-width: 0;
}

```

```

        margin-bottom: 0.5rem;
    }

    .clientes-header .btn-outline-light {
        align-self: center;
        width: fit-content;
    }
}

/* Estilos para botones de acciones en disposición especial */
.acciones-cliente {
    display: flex;
    gap: 0.4rem;
    align-items: flex-start;
    justify-content: center;
    min-width: 140px;
}

.columna-izquierda {
    display: flex;
    flex-direction: column;
    gap: 0.3rem;
}

.columna-derecha {
    display: flex;
    align-items: flex-start;
}

.acciones-cliente .btn {
    font-size: 0.75rem;
    padding: 4px 8px;
    border-radius: 4px;
    font-weight: 600;
    white-space: nowrap;
    min-width: 60px;
}

/* Asegurar que todos los botones tengan el mismo ancho */
.columna-izquierda .btn,
.columna-derecha .btn {
    width: 65px;
}

```

| | └─ Código.css

```
/* Código.css - versión moderna con fondo decorativo y sombras avanzadas */
*/
```

```

:root {
/*
Código.css
-----
- Propósito: estilos para la sección de códigos (lista de códigos, tarjetas, buscador, modal y tablas).
- Estructura: está organizada por secciones claras (variables, layout, cards, tablas, botones, formularios, lista, modal y media queries).
- Uso: mantener las clases/IDs tal como están en el HTML para que los estilos funcionen correctamente.
*/

/* ===== VARIABLES GLOBALES =====
   Definir colores, sombras y constantes de diseño aquí para facilitar la personalización.
*/
--amarillo: #f9c41f;
--negro: #121212;
--blanco: #ffffff;
--gris: #838886;
--gris-claro: #e0e0e0;
--gris-muy-claro: #f5f5f5;
--gris-oscuro: #666666;
--gris-fondo: #f7f7f7;
--naranja: #ff8c00;
--fondo-blanco: rgba(255, 255, 255, 0.95);
--sombra: rgba(0, 0, 0, 0.1);
--sombra-hover: rgba(0, 0, 0, 0.15);
--sombra-moderna: rgba(0, 0, 0, 0.06);
--sombra-elevada: 0 8px 32px rgba(0, 0, 0, 0.12);
--header-h: 72px;
--card-max-w: 1200px;
}

/* === LAYOUT BASE - Modernizado === */
body, html {
  margin: 0;
  /* Contiene reglas base para body/html y fondo decorativo.
     - 'display: flex' permite un sidebar fijo y área de contenido flexible. */
}

body {
  background: var(--gris) !important;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  display: flex;
  min-height: 100vh;
  overflow-x: hidden;
}

```

```

}

/* Fondo decorativo moderno */
body::before,
body::after {
  content: "";
  position: fixed;
  z-index: -1;
  pointer-events: none;
}

body::before {
  top: 0;
  left: 0;
  width: 40%;
  height: 70%;
  background: linear-gradient(135deg, var(--amarillo), #ffd700);
  opacity: 0.9;
}

body::after {
  bottom: 0;
  right: 0;
  width: 60%;
  height: 50%;
  background: linear-gradient(45deg, var(--gris), #a8a8a8);
  opacity: 0.8;
}

/* === CONTENIDO PRINCIPAL - Modernizado === */
.content {
  margin-left: 220px;
  /* Área central donde se coloca la tarjeta principal o el listado de
códigos.
   - margin-left: 220px asume un sidebar; ajustar si cambia la
estructura. */
  padding: 20px 32px;
  flex: 1;
  display: flex;
  align-items: center;
  justify-content: center;
}

/* === CARDS - Modernizadas === */
.card {
  border: 1px solid var(--gris-muy-claro);
  /*
  Sección: Cards

```

```

    - Uso: tarjetas de contenido con borde, sombra y separación.
    - Recomendación: reutilizar para paneles y formularios para mantener
consistencia.

*/
border-radius: 20px;
background: var(--fondo-blanco) !important;
box-shadow: var(--sombra-elevada);
margin-bottom: 24px;
transition: all 0.3s ease;
margin: 0;
overflow: hidden;
max-height: calc(100vh - 6rem);
display: flex;
flex-direction: column;
width: 100%;
backdrop-filter: blur(10px);
position: relative;
}

.card::before {
content: "";
position: absolute;
top: 0;
left: 0;
right: 0;
height: 4px;
background: linear-gradient(90deg, var(--amarillo), var(--naranja));
border-radius: 20px 20px 0 0;
}

.card:hover {
box-shadow: 0 16px 48px rgba(0, 0, 0, 0.15);
transform: translateY(-4px);
}

.card-header {
/* Encabezado de la tarjeta: degradado oscuro y texto destacado */
background: linear-gradient(135deg, var(--negro), #2a2a2a) !important;
color: var(--amarillo) !important;
font-weight: 700;
letter-spacing: 0.8px;
padding: 1.5rem 2rem;
display: flex;
justify-content: space-between;
align-items: center;
position: relative;
overflow: hidden;
}

```

```

.card-header::before {
  content: '';
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background: linear-gradient(45deg,
    transparent 30%,
    rgba(249, 196, 31, 0.05) 50%,
    transparent 70%);
  pointer-events: none;
}

/* === TABLAS - Modernizadas === */
.table {
  background: rgba(255, 255, 255, 0.95);
  border-radius: 16px;
  overflow: hidden;
  border: 1px solid rgba(255, 255, 255, 0.2);
  box-shadow: var(--sombra-elevada);
  backdrop-filter: blur(10px);
  /*
    Sección: Tablas
    - Encabezados con degradado de marca y filas con hover para enfatizar
    registros.
    - position: sticky en th mantiene encabezado visible al hacer scroll.
  */
}

.table th {
  background: linear-gradient(135deg, var(--amarillo), var(--naranja));
  color: var(--negro);
  font-weight: 700;
  border: none;
  padding: 1rem 0.75rem;
  text-shadow: 0 1px 2px rgba(0, 0, 0, 0.1);
  position: sticky;
  top: 0;
  z-index: 10;
  text-transform: uppercase;
  letter-spacing: 0.5px;
  font-size: 0.9rem;
}

.table td {
  border-color: rgba(255, 255, 255, 0.1);
}

```

```

padding: 1rem 0.75rem;
transition: all 0.3s ease;
color: var(--negro) !important; /* Asegura que el texto sea negro */
background: rgba(255, 255, 255, 0.8);
font-weight: 500;
vertical-align: middle;
}

.table tbody tr {
  border-bottom: 1px solid rgba(255, 255, 255, 0.1);
}

.table tbody tr:hover {
  background: linear-gradient(135deg, rgba(249, 196, 31, 0.15), rgba(255, 255, 255, 0.95));
  transform: translateY(-2px);
  box-shadow: 0 6px 20px rgba(249, 196, 31, 0.2);
}

.table tbody tr:hover td {
  background: transparent !important;
  color: var(--negro) !important;
}

.table-responsive {
  width: 100%;
  overflow-x: auto;
  border-radius: 16px;
  scrollbar-width: none; /* Firefox */
  -ms-overflow-style: none; /* IE y Edge */
  /* Contenedor responsive para tablas: habilita scroll-x en pantallas
  pequeñas */
}
.table-responsive::-webkit-scrollbar {
  display: none; /* Chrome, Safari, Opera */
}

.table-responsive table {
  width: 100%;
  min-width: 700px;
  border-collapse: separate;
  border-spacing: 0;
  table-layout: fixed;
}

/* === BOTONES - Modernizados === */
.btn-primary {
  background: linear-gradient(135deg, var(--amarillo), var(--naranja));
}

```

```

color: var(--negro);
border: none;
border-radius: 12px;
padding: 12px 24px;
font-weight: 700;
box-shadow: 0 4px 16px rgba(249, 196, 31, 0.3);
transition: all 0.3s ease;
position: relative;
overflow: hidden;
/* Botón principal con efecto de brillo y estados hover */
}

.btn-primary::before {
  content: "";
  position: absolute;
  top: 0;
  left: -100%;
  width: 100%;
  height: 100%;
  background: linear-gradient(90deg, transparent, rgba(255, 255, 255, 0.2), transparent);
  transition: left 0.5s ease;
}

.btn-primary:hover::before {
  left: 100%;
}

.btn-primary:hover {
  background: linear-gradient(135deg, #e6b619, var(--amarillo));
  transform: translateY(-3px);
  box-shadow: 0 8px 24px rgba(249, 196, 31, 0.4);
}

#btnActualizar {
  background: transparent;
  color: var(--amarillo);
  border: 2px solid var(--amarillo);
  padding: 0.5rem 1rem;
  border-radius: 8px;
  transition: all 0.3s ease;
  font-weight: 600;
  position: relative;
  z-index: 1;
  /* Botón secundario para actualizar listados; estilo outline que cambia en hover */
}

```

```
#btnActualizar:hover {
  background: var(--amarillo);
  color: var(--negro);
  transform: translateY(-2px);
  box-shadow: 0 4px 12px rgba(249, 196, 31, 0.3);
}

/* === CONTENEDORES === */
.container {
  max-width: 102%;
  transform: translateX(1%);
  /* Contenedor general para limitar ancho y alinear contenido */
}

/* === FORMULARIOS - Modernizados === */
.form-control {
  border-radius: 12px;
  border: 2px solid rgba(255, 255, 255, 0.2);
  padding: 14px 16px;
  background: rgba(255, 255, 255, 0.05);
  backdrop-filter: blur(10px);
  color: var(--blanco);
  transition: all 0.3s ease;
  font-weight: 500;
  /* Inputs y controles de formulario
     - Están diseñados para destacar sobre el fondo oscuro con borders
     claros. */
}

.form-control::placeholder {
  color: rgba(255, 255, 255, 0.6);
  font-weight: 400;
}

.form-control:focus {
  border-color: var(--amarillo);
  background: rgba(255, 255, 255, 0.1);
  box-shadow: 0 0 0 4px rgba(249, 196, 31, 0.2), 0 4px 20px rgba(249, 196, 31, 0.15);
  outline: none;
  transform: translateY(-2px);
}

.form-group {
  margin-bottom: 24px;
  position: relative;
}
```

```

.form-label {
  color: var(--blanco);
  font-weight: 600;
  margin-bottom: 8px;
  font-size: 0.95rem;
  text-transform: uppercase;
  letter-spacing: 0.5px;
}

.form-label {
  font-weight: 600;
  color: var(--negro);
  margin-bottom: 0.5rem;
}

/* === LISTA DE CÓDIGOS - Modernizada === */
#codigoLista {
  flex: 1 1 auto;
  overflow-y: auto !important;
  padding: 1rem !important;
  scrollbar-width: none; /* Firefox */
  -ms-overflow-style: none; /* IE y Edge */
  max-height: calc(100vh - 12rem);
  border-radius: 16px;
  background: rgba(255, 255, 255, 0.03);
  backdrop-filter: blur(10px);
  /* Lista principal de códigos
   - Con scroll interno y límites de altura para ocupar la pantalla sin
   desbordar.
   - En mobile se transforma en cards mediante media queries. */
}

#codigoLista::-webkit-scrollbar {
  display: none; /* Chrome, Safari, Opera */
}

#codigoLista th, #codigoLista td {
  text-align: center;
  vertical-align: middle;
  padding: 12px 8px;
  word-wrap: break-word;
  word-break: break-word;
  white-space: normal;
  overflow-wrap: break-word;
  hyphens: auto;
  max-width: 0;
  border-bottom: 1px solid rgba(255, 255, 255, 0.1);
  /* Celdas de la lista: adaptables para mobile, con ruptura de palabra y
  etiquetado por data-label */
}

```

```

}

#codigoLista th {
    background: linear-gradient(135deg, var(--amarillo), var(--naranja));
    color: var(--negro);
    font-weight: 700;
    text-transform: uppercase;
    letter-spacing: 0.5px;
    font-size: 0.85rem;
    position: sticky;
    top: 0;
    z-index: 10;
}

#codigoLista td {
    min-height: 3rem;
    line-height: 1.5;
    color: var(--negro) !important; /* Cambio a texto negro para que se vea */
    transition: all 0.3s ease;
    background: rgba(255, 255, 255, 0.8);
    font-weight: 500;
}

#codigoLista tbody tr:hover td {
    background: rgba(249, 196, 31, 0.15) !important;
    color: var(--negro) !important; /* Mantener texto negro en hover */
}

#buscarCodigo {
    width: 14rem;
    max-width: 100%;
    border: 2px solid rgba(255, 255, 255, 0.2);
    border-radius: 12px;
    background: rgba(255, 255, 255, 0.05);
    color: var(--blanco);
    padding: 12px 16px;
    font-weight: 500;
    backdrop-filter: blur(10px);
    transition: all 0.3s ease;
    font-size: 0.95rem;
}

#buscarCodigo::placeholder {
    color: rgba(255, 255, 255, 0.6);
}

#buscarCodigo:focus {

```

```

border-color: var(--amarillo);
background: rgba(255, 255, 255, 0.1);
box-shadow: 0 0 0 3px rgba(249, 196, 31, 0.2);
outline: none;
}

/* === MODAL CUMPLEAÑOS === */
#modalCumpleanos .modal-content {
  border: 3px solid var(--amarillo) !important;
  border-radius: 15px !important;
  box-shadow: 0 10px 30px rgba(249, 196, 31, 0.3) !important;
  /* Modal estilizado para celebraciones: borde y sombra de marca */
}

#modalCumpleanos .modal-header {
  background: linear-gradient(135deg, var(--amarillo) 0%, #e6b619 100%) !important;
  color: var(--negro) !important;
  border-bottom: 2px solid var(--amarillo) !important;
}

#modalCumpleanos .modal-footer { border-top: 2px solid var(--amarillo) !important; }

#modalCumpleanos .btn-primary {
  background: var(--amarillo) !important;
  border-color: var(--amarillo) !important;
  color: var(--negro) !important;
  font-weight: bold !important;
}

#modalCumpleanos .btn-primary:hover {
  background: #e6b619 !important;
  border-color: #e6b619 !important;
  transform: translateY(-2px) !important;
  box-shadow: 0 4px 12px rgba(249, 196, 31, 0.4) !important;
}

#modalCumpleanos .fas.fa-birthday-cake { animation: bounce 2s infinite; }

@keyframes bounce {
  0%, 20%, 50%, 80%, 100% { transform: translateY(0); }
  40% { transform: translateY(-10px); }
  60% { transform: translateY(-5px); }
}

/* === UTILITY === */
.mover-derecha {

```

```

max-width: 98.5%;
margin-left: 1.6%;
margin-top: 0;
/* Utilidad para desplazar ligeramente contenido a la derecha (ajustes
estéticos) */
}

/* === MEDIA QUERIES - Modernizadas === */
@media (max-width: 992px) {
    .table-responsive table { min-width: 600px; }
    .content { padding: 16px 24px; }
    .card { margin: 0.75rem 0; }
    /* Ajustes de tablet: reducir paddings y tamaño mínimo de tablas */
}

@media (max-width: 768px) {
    body::before {
        width: 60%;
        height: 65%;
        background: radial-gradient(circle, rgba(249, 196, 31, 0.08) 0%,
transparent 70%);
    }
    body::after {
        width: 50%;
        height: 50%;
        background: radial-gradient(circle, rgba(255, 159, 28, 0.06) 0%,
transparent 70%);
    }
}

.content {
    padding: 12px 20px;
    align-items: flex-start;
}

.card {
    margin: 0.75rem 0;
    max-height: calc(100vh - 5rem);
    box-shadow: 0 8px 32px rgba(0, 0, 0, 0.3);
}

#buscarCodigo {
    width: 100%;
    margin-bottom: 1rem;
}

.card-header {
    flex-direction: column;
    gap: 12px;
}

```

```

        text-align: center;
        padding: 20px 24px;
    }
}

@media (max-width: 576px) {
    body::before, body::after { display: none; }

.content {
    align-items: flex-start;
    padding: 8px 12px;
}

.card {
    margin: 0.5rem 0;
    border-left: 4px solid var(--amarillo);
    max-height: calc(100vh - 3rem);
    border-radius: 12px;
}

.table-responsive table {
    min-width: 100%;
    white-space: normal;
    table-layout: auto;
}

#codigoLista th { display: none; }

#codigoLista td {
    display: block;
    text-align: left;
    max-width: none;
    padding: 12px 16px;
    border-bottom: 1px solid rgba(255, 255, 255, 0.1);
}

#codigoLista tr {
    margin-bottom: 1.5rem;
    display: block;
    border: 1px solid rgba(255, 255, 255, 0.15);
    border-radius: 12px;
    background: rgba(255, 255, 255, 0.05);
    backdrop-filter: blur(10px);
    box-shadow: 0 4px 16px rgba(0, 0, 0, 0.2);
}

#codigoLista td::before {
    content: attr(data-label);
}

```

```
    font-weight: 700;
    display: block;
    margin-bottom: 8px;
    color: var(--amarillo);
    text-transform: uppercase;
    letter-spacing: 0.5px;
    font-size: 0.85rem;
  }
}

@media (max-width: 480px) {
  body { overflow-y: auto; }

  .card {
    height: auto;
    max-height: none;
    margin: 0.5rem 0;
  }

  #codigoLista {
    max-height: none;
    overflow-y: visible;
    padding: 0.75rem !important;
  }

  .content {
    padding: 8px;
  }

  .card-header h2 {
    font-size: 1.25rem;
  }
}

/* === RESPONSIVE FINAL ADAPTATIONS === */
@media (max-width: 480px) {
  .form-group,
  .form-control,
  .btn-primary,
  #btnActualizar {
    width: 100%;
    max-width: 100%;
    box-sizing: border-box;
  }

  .form-label {
    font-size: 0.85rem;
  }
}
```

```

.card-header {
  padding: 16px;
  font-size: 1rem;
}

.table-responsive {
  overflow-x: auto;
  -webkit-overflow-scrolling: touch;
}

.table td, .table th {
  font-size: 0.8rem;
  padding: 10px 6px;
}

#modalCumpleanos .modal-dialog {
  margin: 1rem auto;
  max-width: 95%;
}

#modalCumpleanos .modal-content {
  padding: 0.5rem;
}

.btn {
  font-size: 0.9rem;
  padding: 10px;
}

.content {
  padding: 4px;
  flex-direction: column;
}
}

/* === FIX FINAL: Eliminar scroll horizontal en móvil === */
@media (max-width: 480px) {
  html, body {
    overflow-x: hidden !important;
  }

  .content,
  .main-content,
  .table-responsive,
  #codigoLista {
    width: 100% !important;
    max-width: 100% !important;
    overflow-x: hidden !important;
    box-sizing: border-box;
  }
}

```

```
}

.card {
    width: 100% !important;
    box-sizing: border-box;
}

.btn,
.form-control,
#buscarCodigo {
    max-width: 100% !important;
    width: 100% !important;
}

}
```

## | | └ Compra.css

```
/*
Compra.css
-----
- Propósito: estilos específicos para la pantalla de compras (layout, botones, inputs y bloques de interfaz).
- Uso: mantener las clases/IDs tal como están en el HTML para que los estilos y las adaptaciones responsive funcionen.
*/

:root {
    /* Variables globales de color usadas en esta hoja */
    --amarillo: #f9c41f;
    --negro: #000000;
    --blanco: #ffffff;
    --gris: #838886;
}

/* Reset */
html,
body {
    height: 100%;
    margin: 0;
    padding: 0;
    background: var(--blanco);
    color: var(--negro);
    overflow: hidden;
}

/* Layout principal */
.compra-layout {
```

```

/* Contenedor principal que ocupa toda la pantalla y alinea la barra
lateral y el contenido */
display: flex;
height: 100vh;
background: var(--blanco);
}

/* Barra amarilla izquierda */
.compra-barra-amarilla {
    /* Barra decorativa lateral izquierda: color de marca */
    width: 38px;
    background: var(--amarillo);
    box-shadow: 2px 0 12px 0 rgba(0, 0, 0, 0.04);
    z-index: 10;
}

/* Contenedor principal con fondo animado */
.template-container {
    /* Contenedor flexible que tiene un fondo oscuro animado y contiene el
bloque principal */
    flex: 1;
    position: relative;
    overflow: hidden;
    background: linear-gradient(
        135deg,
        var(--negro) 0%,
        #1a1a1a 25%,
        #2a2a2a 50%,
        #1a1a1a 75%,
        var(--negro) 100%
    );
}

.template-container::before {
    content: "";
    position: absolute;
    inset: 0;
    background: linear-gradient(
        45deg,
        transparent 30%,
        rgba(249, 196, 31, 0.03) 50%,
        transparent 70%
    );
    animation: diagonal-sweep 8s infinite;
}

@keyframes diagonal-sweep {
    0%,

```

```

    100% {
      transform: translateX(-100%) translateY(-100%);
    }
    50% {
      transform: translateX(100%) translateY(100%);
    }
  }

/* Bloque principal */
.compra-bloque-inicio {
  /* Bloque principal centrado dentro del template-container
   - Contiene el formulario/tabla principal de compras.
   - max-height y max-width limitan el tamaño para mantener la UI
  contenida. */
  width: 96%;
  max-width: 2500px;
  height: 90%;
  max-height: 1800px;
  background: var(--blanco);
  box-shadow: 0 2px 18px 0 rgba(40, 40, 40, 0.09);
  border-radius: 2.5em;
  margin: 4vh auto 0; /* Centra horizontalmente y da margen superior */
  padding: 4% 3%;
  z-index: 2;
  display: flex;
  flex-direction: column;
  opacity: 0.98;
  justify-content: flex-start;
  position: relative; /* Cambio clave */
}

/* Header en una línea */
.compra-header-linea {
  /* Header compacto que agrupa título, opciones y buscador en una línea */
  display: flex;
  align-items: center;
  gap: 1.2em;
  justify-content: flex-start;
  width: 100%;
  margin-bottom: 4%;
  flex-wrap: wrap;
}

/* Título */
.compra-titulo {
  /* Título del bloque de compras, estilo uppercase y destacado */
  font-size: 1.3em;
}

```

```
font-weight: bold;
letter-spacing: 1px;
text-transform: uppercase;
color: var(--negro);
min-width: 110px;
margin-right: 0.4em;
}

/* Botones de opción */
.compra-btn-opcion {
    /* Botones de filtro/selección con estilo pill y estado :hover /
.selected */
    background: #f9fafb;
    color: var(--negro);
    border: 1.5px solid var(--amarillo);
    border-radius: 1.3em;
    padding: 0.7em 1.5em;
    font-weight: bold;
    font-size: 0.5em;
    cursor: pointer;
    transition: background 0.18s, color 0.18s;
    text-transform: uppercase;
    letter-spacing: 1.2px;
    min-width: 90px;
    max-width: 170px;
    margin-left: -2em;
}
.compra-btn-opcion:hover,
.compra-btn-opcion.selected {
    background: var(--amarillo);
}

/* Input y botón de búsqueda */
.compra-input-id {
    /* Input para ingresar ID o número de orden
       - Estilo compacto y sin flechas de número para mejor experiencia
móvil. */
    border: 1.5px solid #bcdbc;
    border-radius: 1em;
    padding: 0.5em 1.1em;
    font-size: 1em;
    max-width: 160px;
    min-width: 60px;
    outline: none;
    background: #f9fafb;
    transition: border-color 0.2s;
    margin-left: 0;
    display: block;
```

```

}

/* Ocultar flechas del input number sin warnings */
.compra-input-id::-webkit-outer-spin-button,
.compra-input-id::-webkit-inner-spin-button {
    appearance: none;
    -webkit-appearance: none;
    margin: 0;
}
.compra-input-id:focus {
    border-color: var(--amarillo);
}

.compra-input-container {
    position: relative;
    display: inline-block;
    width: auto;
    vertical-align: top;
    margin-left: auto;
}

.compra-btn-buscar {
    background: var(--amarillo);
    color: var(--negro);
    border: none;
    border-radius: 1em;
    padding: 0.5em 1.1em;
    font-weight: bold;
    cursor: pointer;
    transition: background 0.2s;
    letter-spacing: 1px;
    box-shadow: 0 1px 3px rgba(0, 0, 0, 0.06);
    font-size: 1em;
    margin-left: 0.5em;
    display: block;
}
.compra-btn-buscar:hover {
    background: #ffdc52;
}

/* Indicación debajo */
/* Elimina una de las dos reglas duplicadas y deja solo esta: */
.compra-indicacion {
    /* Mensaje informativo que aparece centrado si no hay resultados o para
instrucciones */
    text-align: center;
    font-size: 1em;
    color: #666;
}

```

```

background: #f7f7f7e0;
border-radius: 1em;
padding: 1.1em 0.5em;
margin-top: 8em;
opacity: 0.98;
display: flex;
justify-content: center;
align-items: center;
gap: 0.7em;
}

/* Barra derecha */
.compra-barra-derecha {
    /* Barra lateral derecha fija (decorativa o para accesos rápidos) */
    position: fixed;
    top: 0;
    right: 0;
    width: 20px;
    height: 100vh;
    background: var(--blanco);
    z-index: 9;
    box-shadow: -1px 0 6px 0 rgba(0, 0, 0, 0.04);
}

.compra-indicacion .bi-info-circle {
    color: var(--amarillo);
    font-size: 1.3em;
}

.card.shadow.mb-4 {
    /* Tarjeta con sombra y borde izquierdo amarillo para resaltar secciones importantes */
    background: var(--blanco);
    box-shadow: 0 0 10px rgba(249, 196, 31, 0.12);
    border-left: 5px solid var(--amarillo);
    border-radius: 1em;
    margin-top: -1.5em;
    margin-bottom: 0.5em; /* Muy reducido */
    height: 72%;
    overflow-y: visible;
}

.card-body {
    /* Card body compacto para forms y listados */
    padding: 0.5em 1.2em; /* Más compacto en todos los lados */
    line-height: 1.3; /* Reduce altura de líneas de texto */
}

```

```

/* Compactar elementos del formulario */
.card-body .form-group,
.card-body .mb-3,
.card-body .mb-2 {
  margin-bottom: 0.75rem !important; /* Reduce espacios entre campos */
}

.form-control:read-only {
  background: #f9fafb;
  color: #888;
  padding: 0.4rem 0.75rem; /* Campos más compactos */
}

/* ---- Estilo especial para el botón "Acumular" (cuando está activo) --- */
.btn-acumular {
  /* Botón con estilo de éxito para acciones de acumulación */
  background: #28a745 !important; /* Verde Bootstrap */
  color: #fff !important;
  border: none;
  box-shadow: 0 2px 7px 0 rgba(40, 167, 69, 0.1);
  transition: background 0.2s, color 0.2s;
  width: 110px;
}

.mensaje-error {
  /* Mensaje de error oculto por defecto; mostrar añadiendo la clase
   .show */
  display: none;
  color: #d43b3b;
  font-weight: bold;
  font-size: 0.95em;
  display: flex;
}

/* Clase para mostrar el mensaje cuando sea necesario */
.mensaje-error.show {
  display: block;
}

/* Estilos para el botón con ícono de búsqueda */
.compra-btn-buscar-icon {
  /* Botón circular superpuesto al input para acciones rápidas */
  position: absolute;
  right: 8px;
  top: 50%;
  transform: translateY(-50%);
  background: var(--amarillo);
}

```

```
border: none;
border-radius: 4px;
width: 32px;
height: 32px;
display: flex;
align-items: center;
justify-content: center;
cursor: pointer;
transition: all 0.3s ease;
z-index: 2;
}

.compra-btn-buscar-icon:hover {
  background: #f0c419;
  transform: translateY(-50%) scale(1.1);
}

.compra-btn-buscar-icon i {
  color: var(--negro);
  font-size: 14px;
}

/* Ajustar el padding del input para dar espacio al ícono */
.compra-input-container .compra-input-id {
  padding-right: 45px;
}

/* ===== RESPONSIVE ===== */

/* Para pantallas menores o iguales a 768px (móvil) */
@media (max-width: 768px) {
  /* Ajustes para móviles: convertir layout a columna y adaptar tamaños */
  html,
  body {
    overflow-y: auto;
  }

  .compra-layout {
    flex-direction: column;
    height: auto;
    min-height: 100vh;
  }

  .compra-barra-amarilla {
    width: 100%;
    height: 10px;
    box-shadow: 0 2px 6px rgba(0, 0, 0, 0.04);
  }
}
```

```
}

.template-container {
  flex: 1;
  padding-bottom: 2em;
  min-height: 100vh;
}

.compra-bloque-inicio {
  position: static;
  width: 92%;
  min-height: 83vh; /* ← Aquí se aumenta la altura mínima */
  max-height: none;
  margin: 5vh auto;
  padding: 6% 4%;
}

.compra-header-linea {
  flex-direction: column;
  align-items: flex-start;
  gap: 0.8em;
}

.compra-titulo {
  font-size: 1.1em;
  margin-right: 0;
}

.compra-btn-opcion {
  margin-left: 0;
  font-size: 0.65em;
  padding: 0.6em 1.2em;
  width: 100%;
  max-width: none;
}

.compra-input-id {
  max-width: 100%;
  width: 100%;
  margin-left: 0;
  font-size: 0.95em;
}

.compra-btn-buscar {
  margin-left: 0;
  width: 100%;
  margin-top: 0.8em;
}
```

```
.compra-input-container {
  width: 100%;
  margin-left: 0;
}

.compra-indicacion {
  font-size: 0.95em;
  padding: 0.8em;
  margin-top: 4em;
  flex-direction: column;
  gap: 0.5em;
  text-align: center;
}

.compra-barra-derecha {
  display: none;
}

.card.shadow.mb-4 {
  height: auto;
  margin-top: 1em;
}

.btn-acumular {
  width: 100%;
}

/* Para tablets y pantallas medianas */
@media (min-width: 769px) and (max-width: 1024px) {
  .compra-bloque-inicio {
    max-width: 95%;
    height: auto;
  }

  .compra-header-linea {
    flex-wrap: wrap;
    gap: 1em;
  }

  .compra-input-id {
    margin-left: 0;
  }

  .compra-btn-buscar {
    margin-left: 0;
  }
}
```

```

.compra-input-container {
  margin-left: 0;
}
}

| | └── Cumple.css
/*
Cumple.css
-----
- Propósito: estilos para la sección de "Cumpleaños" (listados, tarjetas, formularios y envíos de correo).
- Estructura: organizado por bloques: variables, layout, fondos, cards, botones, alerts, tablas, formularios, badges, elementos especiales, animaciones y responsive.
- Uso: conservar las clases e IDs en el HTML para que los estilos y comportamientos responsive funcionen correctamente.
*/

/* ===== VARIABLES GLOBALES =====
   Definir colores, variantes y sombras aquí para facilitar cambios de tema.
*/
:root {
  /* === Colores principales === */
  --amarillo: #f9c41f;
  --gris: #838886;
  --negro: #121212;
  --blanco: #fff;
  --gris-fondo: #f7f7f7;
  --gris-borde: #e0e0e0;
  --success: #28a745;
  --warning: #ffc107;
  --danger: #dc3545;
  --info: #17a2b8;

  /* === Colores extendidos === */
  --naranja: #ff9f1c;
  --gris-claro: #f8f9fa;
  --gris-muy-claro: #e9ecef;
  --azul-cielo: #87ceeb;
  --rosa-suave: #ffb6c1;
  --verde-menta: #98fb98;
  --fondo-blanco: rgba(255, 255, 255, 0.95);

  /* === Sistema de sombras === */
  --sombra: rgba(0, 0, 0, 0.1);
  --sombra-hover: rgba(0, 0, 0, 0.15);
}

```

```

--sombra-moderna: rgba(0, 0, 0, 0.06);
--sombra-elevada: 0 8px 32px rgba(0, 0, 0, 0.12);
--sombra-suave: 0 2px 8px rgba(0, 0, 0, 0.08);
}

* {
  box-sizing: border-box;
}

/* === LAYOUT BASE - Modernizado === */
body {
  /* Layout principal: se usa display:flex para permitir sidebar fijo y
  contenido flexible.
    - background y overflow-x están ajustados para evitar scroll
  horizontal inesperado.
    - Usar con cuidado los !important si se necesita sobrescribir en
  otras hojas. */
  margin: 0 !important;
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif !important;
  display: flex !important;
  min-height: 100vh !important;
  background: var(--gris) !important;
  overflow-x: hidden !important;
  position: relative;
}

body::-webkit-scrollbar {
  display: none !important;
}

/* Fondos decorativos modernos */
body::before,
body::after {
  /* Pseudo-elementos que añaden fondos decorativos en la página.
    - Se usan para dar profundidad y color sin añadir elementos HTML.
    - pointer-events: none para no interferir con la interacción. */
  content: "";
  position: fixed;
  z-index: -1 !important;
  pointer-events: none;
}

body::before {
  top: 0;
  left: 0;
  width: 45%;
  height: 75%;
}

```

```

background: radial-gradient(circle at 30% 30%, rgba(249, 196, 31, 0.15)
0%, transparent 50%),
            linear-gradient(135deg, var(--amarillo) 0%, var(--naranja)
100%);
border-radius: 0 0 50% 0;
filter: blur(1px);
}

body::after {
bottom: 0;
right: 0;
width: 65%;
height: 55%;
background: radial-gradient(circle at 70% 70%, rgba(255, 159, 28, 0.12)
0%, transparent 50%),
            linear-gradient(225deg, var(--gris) 0%, #6c757d 100%);
border-radius: 50% 0 0 0;
filter: blur(1px);
}
.content {
/* Área principal donde se renderizan tarjetas y listas.
   - margin-left: 220px asume un sidebar; ajustar si la estructura del
layout cambia. */
margin-left: 220px !important;
padding: 32px !important;
flex: 1 !important;
position: relative;
z-index: 1;
}

.container {
max-width: 100% !important;
padding: 0;
}

/* === CARDS - Modernizadas === */
.card {
/* Tarjetas: fondo semitransparente, bordes redondeados y sombra
elevada.
   - Reutilizar .card para paneles y formularios para consistencia
visual. */
border: none !important;
border-radius: 20px !important;
background: var(--fondo-blanco) !important;
overflow: hidden !important;
box-shadow: var(--sombra-elevada) !important;
margin-bottom: 32px !important;
transition: all 0.3s ease !important;
}

```

```
backdrop-filter: blur(10px);
position: relative;
}

.card::before {
content: "";
position: absolute;
top: 0;
left: 0;
width: 100%;
height: 4px;
background: linear-gradient(90deg, var(--amarillo) 0%, var(--naranja) 100%);
z-index: 1;
}

.card:hover {
transform: translateY(-8px);
box-shadow: 0 16px 48px rgba(0, 0, 0, 0.2) !important;
}

.card.shadow.mb-4 {
box-shadow: var(--sombra-elevada) !important;
border: 1px solid rgba(249, 196, 31, 0.2) !important;
position: relative;
overflow: hidden;
}

.card.shadow.mb-4::before {
height: 6px;
background: linear-gradient(90deg, var(--amarillo) 0%, var(--naranja) 50%, #ffd700 100%);
}

.card-header {
/* Encabezado de tarjeta: degradado oscuro y texto destacado en amarillo.
- Incluye iconos con efecto de sombra para dar énfasis. */
background: linear-gradient(135deg, var(--negro) 0%, #2c2c2c 100%) !important;
color: var(--blanco) !important;
font-weight: 700 !important;
letter-spacing: 0.8px !important;
padding: 24px 28px !important;
border-bottom: none !important;
display: flex !important;
justify-content: space-between !important;
align-items: center !important;
```

```

gap: 12px;
position: relative;
z-index: 2;
}

.card-header h5,
.card-header h6 {
margin: 0;
font-size: 1.25rem;
color: var(--amarillo) !important;
text-transform: uppercase;
text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);
font-weight: 700;
}

.card-header i {
color: var(--amarillo) !important;
font-size: 1.4rem;
text-shadow: 0 2px 8px rgba(249, 196, 31, 0.5);
}

/* Header principal uniformado - Modernizado */
.card-header.bg-dark.text-white {
padding: 24px 28px !important;
min-height: 70px;
background: linear-gradient(135deg, var(--negro) 0%, #2c2c2c 100%) !important;
}

@media (max-width: 992px) {
.card-header.bg-dark.text-white {
padding: 20px 24px !important;
min-height: 65px;
}
}

@media (max-width: 768px) {
.card-header.bg-dark.text-white {
padding: 16px 20px !important;
min-height: 60px;
}
}

/* === BOTONES - Modernizados === */
.btn {
/* Botones globales: estilo moderno con animación de brillo en pseudo-elemento */
border: none !important;
border-radius: 12px !important;
}

```

```
font-weight: 700 !important;
transition: all 0.3s ease !important;
box-shadow: 0 4px 16px rgba(0, 0, 0, 0.1) !important;
position: relative;
overflow: hidden;
text-transform: uppercase;
letter-spacing: 0.5px;
padding: 12px 24px !important;
}

.btn::before {
  content: "";
  position: absolute;
  top: 0;
  left: -100%;
  width: 100%;
  height: 100%;
  background: linear-gradient(90deg, transparent, rgba(255, 255, 255, 0.2), transparent);
  transition: left 0.5s ease;
}

.btn:hover::before {
  left: 100%;
}

.btn-primary,
.btn-warning {
  background: linear-gradient(135deg, var(--amarillo), var(--naranja)) !important;
  color: var(--negro) !important;
  box-shadow: 0 4px 16px rgba(249, 196, 31, 0.3) !important;
}

.btn-primary:hover,
.btn-warning:hover {
  background: linear-gradient(135deg, #e6b619, var(--amarillo)) !important;
  transform: translateY(-3px) !important;
  box-shadow: 0 8px 24px rgba(249, 196, 31, 0.4) !important;
}

.btn-success {
  background: linear-gradient(135deg, var(--success), #20c997) !important;
  color: var(--blanco) !important;
  box-shadow: 0 4px 16px rgba(40, 167, 69, 0.3) !important;
}
```

```
.btn-success:hover {
  background: linear-gradient(135deg, #218838, var(--success)) !important;
  transform: translateY(-3px) !important;
  box-shadow: 0 8px 24px rgba(40, 167, 69, 0.4) !important;
}

.btn-danger {
  background: linear-gradient(135deg, var(--danger), #e55565) !important;
  color: var(--blanco) !important;
  box-shadow: 0 4px 16px rgba(220, 53, 69, 0.3) !important;
}

.btn-danger:hover {
  background: linear-gradient(135deg, #c82333, var(--danger)) !important;
  transform: translateY(-3px) !important;
  box-shadow: 0 8px 24px rgba(220, 53, 69, 0.4) !important;
}

.btn-sm {
  padding: 8px 16px !important;
  font-size: 0.85rem !important;
  border-radius: 8px !important;
}
/* === ALERTS - Modernizadas === */
.alert {
  /* Alertas con fondo y borde suave; pseudo-elemento superior para acento */
  border: none !important;
  border-radius: 16px !important;
  box-shadow: var(--sombra-suave) !important;
  margin-bottom: 24px !important;
  padding: 20px 24px !important;
  position: relative;
  overflow: hidden;
  backdrop-filter: blur(10px);
}

.alert::before {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 4px;
  z-index: 1;
}
```

```

.alert-info {
  background: rgba(232, 244, 248, 0.9) !important;
  color: #0c5460 !important;
  border: 1px solid rgba(23, 162, 184, 0.2) !important;
}

.alert-info::before {
  background: linear-gradient(90deg, var(--info), #20c997);
}

.alert-warning {
  background: rgba(255, 243, 205, 0.9) !important;
  color: #856404 !important;
  border: 1px solid rgba(255, 193, 7, 0.2) !important;
}

.alert-warning::before {
  background: linear-gradient(90deg, var(--warning), var(--amarillo));
}

.alert-success {
  background: rgba(212, 237, 218, 0.9) !important;
  color: #155724 !important;
  border: 1px solid rgba(40, 167, 69, 0.2) !important;
}

.alert-success::before {
  background: linear-gradient(90deg, var(--success), #20c997);
}

/* === TABLAS - Modernizadas === */
.table,
.table-bordered,
.table-hover {
  /* Tablas estilizadas: fondo claro, bordes suaves y headers sticky para scroll */
  background: var(--fondo-blanco) !important;
  border-radius: 16px !important;
  overflow: hidden !important;
  box-shadow: var(--sombra-elevada) !important;
  border: none !important;
  border-collapse: separate !important;
  border-spacing: 0 !important;
  backdrop-filter: blur(10px);
}

.table thead th {

```

```

background: linear-gradient(135deg, var(--negro), #2c2c2c) !important;
color: var(--blanco) !important;
font-weight: 700 !important;
padding: 20px 16px !important;
font-size: 0.9rem !important;
text-align: center !important;
border: none !important;
text-transform: uppercase;
letter-spacing: 0.8px;
position: sticky;
top: 0;
z-index: 10;
}
.table tbody td {
padding: 16px 15px !important;
border-top: 1px solid rgba(255, 255, 255, 0.1) !important;
font-size: 0.95rem !important;
vertical-align: middle !important;
text-align: center !important;
border: none !important;
color: var(--negro) !important;
font-weight: 500;
transition: all 0.3s ease;
background: rgba(255, 255, 255, 0.8);
}
.table tbody tr {
transition: all 0.3s ease;
border-bottom: 1px solid rgba(255, 255, 255, 0.1);
}
.table tbody tr:hover {
transform: translateY(-2px);
box-shadow: 0 4px 16px rgba(249, 196, 31, 0.15);
}
.table tbody tr:hover td {
background: rgba(249, 196, 31, 0.1) !important;
color: var(--negro) !important;
}
.table tbody tr:nth-child(even) td {
background: rgba(255, 255, 255, 0.6);
}
.table tbody tr:nth-child(even):hover td {
background: rgba(249, 196, 31, 0.12) !important;
}

```

```

/* Contenedor de tabla con scroll horizontal modernizado */
.table-responsive {
    /* Contenedor responsive para tablas: permite scroll-x en dispositivos
    pequeños y custom scrollbars */
    border-radius: 16px !important;
    overflow-x: auto !important;
    overflow-y: hidden !important;
    -webkit-overflow-scrolling: touch;
    scrollbar-width: none;
} /* Firefox */
.table-responsive::-webkit-scrollbar {
    width: 8px;
    height: 8px;
} /* WebKit */

.table-responsive::-webkit-scrollbar-track {
    background: rgba(255, 255, 255, 0.1);
    border-radius: 8px;
}

.table-responsive::-webkit-scrollbar-thumb {
    background: linear-gradient(135deg, var(--amarillo), var(--naranja));
    border-radius: 8px;
    box-shadow: 0 2px 8px rgba(249, 196, 31, 0.3);
}

.table-responsive::-webkit-scrollbar-thumb:hover {
    background: linear-gradient(135deg, #e6b619, var(--amarillo));
}

/* === FORMULARIOS - Modernizados === */
.form-control {
    /* Inputs del formulario: bordes visibles y foco destacado para
    accesibilidad */
    border-radius: 12px !important;
    border: 2px solid rgba(255, 255, 255, 0.3) !important;
    padding: 14px 16px !important;
    font-size: 0.95rem !important;
    transition: all 0.3s ease !important;
    background: rgba(255, 255, 255, 0.9) !important;
    color: var(--negro) !important;
    font-weight: 500;
    backdrop-filter: blur(10px);
}

.form-control::placeholder {
    color: rgba(0, 0, 0, 0.4);
}

```

```

    font-weight: 400;
    font-style: italic;
}

.form-control:focus {
    border-color: var(--amarillo) !important;
    background: rgba(255, 255, 255, 0.95) !important;
    box-shadow: 0 0 0 4px rgba(249, 196, 31, 0.2), 0 4px 20px rgba(249, 196, 31, 0.15) !important;
    transform: translateY(-2px);
}

/* === CAMPOS VACÍOS - Más visibles === */
.form-control:not(:focus):placeholder-shown,
.form-control:not(:focus):invalid,
.form-control:not(:focus):empty,
.form-control:not(:focus)[value=""],
input[type="text"]:not(:focus):placeholder-shown,
input[type="email"]:not(:focus):placeholder-shown,
input[type="tel"]:not(:focus):placeholder-shown,
textarea:not(:focus):placeholder-shown,
textarea:not(:focus):empty,
textarea:not(:focus)[value=""] {
    background: rgba(249, 196, 31, 0.05) !important;
    border-color: rgba(249, 196, 31, 0.3) !important;
    box-shadow: inset 0 2px 4px rgba(249, 196, 31, 0.1) !important;
}
/* Estilos para campos vacíos: resaltan visualmente para indicar que se deben completar */

.form-control:not(:focus):placeholder-shown:hover,
.form-control:not(:focus):invalid:hover,
.form-control:not(:focus):empty:hover,
.form-control:not(:focus)[value=""]:hover,
input[type="text"]:not(:focus):placeholder-shown:hover,
input[type="email"]:not(:focus):placeholder-shown:hover,
input[type="tel"]:not(:focus):placeholder-shown:hover,
textarea:not(:focus):placeholder-shown:hover,
textarea:not(:focus):empty:hover,
textarea:not(:focus)[value=""]:hover {
    background: rgba(249, 196, 31, 0.08) !important;
    border-color: rgba(249, 196, 31, 0.4) !important;
    box-shadow: 0 2px 8px rgba(249, 196, 31, 0.15) !important;
}

.form-control-sm {
    padding: 10px 14px !important;
    font-size: 0.875rem !important;
}

```

```

    border-radius: 10px !important;
}

.form-label {
    font-weight: 700 !important;
    color: var(--negro) !important;
    margin-bottom: 8px !important;
    font-size: 0.95rem !important;
    text-transform: uppercase;
    letter-spacing: 0.5px;
}

/* === FORMULARIO DE CORREO DE FELICITACIÓN - Mejorado === */
/* Contraste específico para formularios de felicitación */
.card-body form input,
.card-body form textarea,
.modal form input,
.modal form textarea {
    border-radius: 12px !important;
    border: 2px solid rgba(255, 255, 255, 0.3) !important;
    padding: 14px 16px !important;
    font-size: 0.95rem !important;
    transition: all 0.3s ease !important;
    background: rgba(255, 255, 255, 0.9) !important;
    color: var(--negro) !important;
    font-weight: 500;
    backdrop-filter: blur(10px);
    width: 100%;
    box-sizing: border-box;
}

/* Placeholder mejorado para todos los inputs */
.card-body form input::placeholder,
.card-body form textarea::placeholder,
.modal form input::placeholder,
.modal form textarea::placeholder {
    color: rgba(0, 0, 0, 0.4);
    font-weight: 400;
    font-style: italic;
}

/* Campos vacíos en formularios de felicitación */
.card-body form input:not(:focus):placeholder-shown,
.card-body form input:not(:focus):invalid,
.card-body form input:not(:focus)[value=""],
.card-body form textarea:not(:focus):placeholder-shown,
.card-body form textarea:not(:focus):empty,
.card-body form textarea:not(:focus)[value=""],

```

```

.modal form input:not(:focus):placeholder-shown,
.modal form input:not(:focus):invalid,
.modal form input:not(:focus)[value=""],
.modal form textarea:not(:focus):placeholder-shown,
.modal form textarea:not(:focus):empty,
.modal form textarea:not(:focus)[value=""] {
  background: rgba(249, 196, 31, 0.05) !important;
  border-color: rgba(249, 196, 31, 0.3) !important;
  box-shadow: inset 0 2px 4px rgba(249, 196, 31, 0.1) !important;
}

/* Hover en campos vacíos de formularios */
.card-body form input:not(:focus):placeholder-shown:hover,
.card-body form input:not(:focus):invalid:hover,
.card-body form input:not(:focus)[value=""]:hover,
.card-body form textarea:not(:focus):placeholder-shown:hover,
.card-body form textarea:not(:focus):empty:hover,
.card-body form textarea:not(:focus)[value=""]:hover,
.modal form input:not(:focus):placeholder-shown:hover,
.modal form input:not(:focus):invalid:hover,
.modal form input:not(:focus)[value=""]:hover,
.modal form textarea:not(:focus):placeholder-shown:hover,
.modal form textarea:not(:focus):empty:hover,
.modal form textarea:not(:focus)[value=""]:hover {
  background: rgba(249, 196, 31, 0.08) !important;
  border-color: rgba(249, 196, 31, 0.4) !important;
  box-shadow: 0 2px 8px rgba(249, 196, 31, 0.15) !important;
  transform: translateY(-1px);
}

/* Focus mejorado para formularios */
.card-body form input:focus,
.card-body form textarea:focus,
.modal form input:focus,
.modal form textarea:focus {
  border-color: var(--amarillo) !important;
  background: rgba(255, 255, 255, 0.95) !important;
  box-shadow: 0 0 0 4px rgba(249, 196, 31, 0.2), 0 4px 20px rgba(249, 196, 31, 0.15) !important;
  transform: translateY(-2px);
  outline: none;
}

/* === MEJORAS ESPECÍFICAS PARA ENVÍO DE CORREO === */
/* Contenedor del formulario de correo con mejor contraste */
.card-body {
  /* Card-body: usado como contenedor para formularios de felicitación,
  con overlay util en hover */
}

```

```

    position: relative;
}
td.correo {
    white-space: normal;
    word-break: break-all;
}
td.contacto {
    white-space: normal;
    word-break: break-word;
}

.card-body::before {
    content: "";
    position: absolute;
    top: -2px;
    left: -2px;
    right: -2px;
    bottom: -2px;
    background: linear-gradient(45deg, transparent, rgba(249, 196, 31, 0.1), transparent);
    border-radius: 20px;
    z-index: -1;
    opacity: 0;
    transition: opacity 0.3s ease;
}

.card:hover .card-body::before {
    opacity: 1;
}

/* Efectos de pulsación sutil para campos vacíos */
.card-body form input:not(:focus):placeholder-shown,
.card-body form textarea:not(:focus):placeholder-shown,
.card-body form input:not(:focus):empty,
.card-body form textarea:not(:focus):empty {
    animation: subtlePulse 3s ease-in-out infinite;
}

@keyframes subtlePulse {
    0%, 100% {
        box-shadow: inset 0 2px 4px rgba(249, 196, 31, 0.1);
    }
    50% {
        box-shadow: inset 0 2px 4px rgba(249, 196, 31, 0.15), 0 0 0 1px
                    rgba(249, 196, 31, 0.1);
    }
}

```

```

/* Indicador visual para campos que se llenarán automáticamente */
.card-body form input:not(:focus):placeholder-shown::after,
.card-body form textarea:not(:focus):placeholder-shown::after {
    content: "✉";
    position: absolute;
    right: 12px;
    top: 50%;
    transform: translateY(-50%);
    opacity: 0.3;
    font-size: 16px;
    pointer-events: none;
}

/* Mejora para labels del formulario de correo */
.card-body .form-label {
    position: relative;
    display: inline-block;
}

.card-body .form-label::after {
    content: "";
    position: absolute;
    bottom: -2px;
    left: 0;
    width: 0;
    height: 2px;
    background: linear-gradient(90deg, var(--amarillo), var(--naranja));
    transition: width 0.3s ease;
}

.card-body .form-label:hover::after {
    width: 100%;
}

/* === BADGES - Modernizados === */
:is(
    .cumple-badge,
    .table .badge,
    #cumpleLista .badge,
    #historialCumples .badge
) {
    padding: 8px 16px !important;
    border-radius: 25px !important;
    font-size: 0.8rem !important;
    font-weight: 700 !important;
    display: inline-block !important;
    text-transform: uppercase;
}

```

```

letter-spacing: 0.5px;
box-shadow: 0 2px 8px rgba(0, 0, 0, 0.15);
transition: all 0.3s ease;
}

:is(.bg-danger) {
background: linear-gradient(135deg, var(--danger), #e55565) !important;
color: var(--blanco) !important;
box-shadow: 0 4px 12px rgba(220, 53, 69, 0.3) !important;
}

:is(.bg-success) {
background: linear-gradient(135deg, var(--success), #20c997)
!important;
color: var(--blanco) !important;
box-shadow: 0 4px 12px rgba(40, 167, 69, 0.3) !important;
}

:is(.bg-warning) {
background: linear-gradient(135deg, var(--warning), var(--amarillo))
!important;
color: var(--negro) !important;
box-shadow: 0 4px 12px rgba(255, 193, 7, 0.3) !important;
}

/* === ELEMENTOS ESPECIALES - Modernizados === */
#rangoSemana {
    /* Elemento destacado para seleccionar rango de semana con efecto de
brillo al hover */
    background: linear-gradient(135deg, var(--amarillo), var(--naranja))
!important;
    color: var(--negro) !important;
    font-weight: 700 !important;
    border: none;
    border-radius: 16px !important;
    box-shadow: 0 6px 20px rgba(249, 196, 31, 0.4) !important;
    padding: 12px 20px !important;
    text-transform: uppercase;
    letter-spacing: 0.8px;
    position: relative;
    overflow: hidden;
}

#rangoSemana::before {
    content: "";
    position: absolute;
    top: 0;
    left: -100%;
}

```

```

width: 100%;
height: 100%;
background: linear-gradient(90deg, transparent, rgba(255, 255, 255, 0.3), transparent);
transition: left 0.6s ease;
}

#rangoSemana:hover::before {
left: 100%;
}

.spinner-border.text-warning {
color: var(--amarillo) !important;
border-color: rgba(249, 196, 31, 0.2) !important;
border-top-color: var(--amarillo) !important;
border-width: 3px !important;
animation: spin 1s linear infinite, pulse 2s ease-in-out infinite alternate;
}

@keyframes pulse {
0% { transform: scale(1); }
100% { transform: scale(1.1); }
}

/* === ANIMACIONES - Mejoradas === */
#recordatorioLlamadas {
/* Animación fadeInUp para aparecer contenido con suavidad */
animation: fadeInUp 0.6s ease-out !important;
}

@keyframes fadeInUp {
from {
    opacity: 0;
    transform: translateY(30px);
}
to {
    opacity: 1;
    transform: translateY(0);
}
}

/* === OVERRIDES INLINE - Modernizados === */
#cumpleLista [style*="background"],
#historialCumples [style*="background"],
.table [style*="background"] {
background: var(--fondo-blanco) !important;
color: var(--negro) !important;
}

```

```

}

#cumpleLista th[style*="background"],
#historialCumples th[style*="background"],
.table th[style*="background"] {
  background: var(--negro) !important;
  color: var(--blanco) !important;
}

.table .btn {
  padding: 0.25rem 0.5rem !important;
  font-size: 0.75rem !important;
  line-height: 1.5 !important;
  border-radius: 0.2rem !important;
}

.table .btn-warning {
  background: var(--amarillo) !important;
  border-color: var(--amarillo) !important;
  color: var(--negro) !important;
}

.table .btn-warning:hover {
  background: #e0a800 !important;
  border-color: #d39e00 !important;
  color: var(--negro) !important;
}

#historialCumples .table tbody tr:nth-child(even) td {
  background: #f8f9fa !important;
}

.btn-nav-week{
  border: none;
  outline: none;
  cursor: pointer;
  padding: 6px 10px;
  border-radius: 9999px;
  background: #F9C41F;
  box-shadow: 0 2px 8px rgba(0,0,0,.15);
}

.btn-nav-week:disabled{
  opacity: .5;
  cursor: not-allowed;
}

}

/* === RESPONSIVE BREAKPOINTS - Modernizados === */
@media (max-width: 992px) {
  /* Breakpoints: ajustes para tablet y móvil grande */
  .content {
    margin-left: 0 !important;
}

```

```
padding: 24px 20px !important;
}

.container {
  max-width: 98% !important;
  margin: 0 auto !important;
  transform: none !important;
}

.card {
  margin-bottom: 20px !important;
  border-radius: 16px !important;
}

.card-header {
  padding: 20px 24px !important;
}

.card-header h5,
.card-header h6 {
  font-size: 1.1rem;
}

body::before {
  width: 50%;
  height: 65%;
}

body::after {
  width: 70%;
  height: 45%;
}

}

@media (max-width: 768px) {
  body::before {
    width: 60%;
    height: 50%;
    background: radial-gradient(circle, rgba(249, 196, 31, 0.12) 0%, transparent 70%);
  }

  body::after {
    width: 55%;
    height: 40%;
    background: radial-gradient(circle, rgba(131, 136, 134, 0.08) 0%, transparent 70%);
  }
}
```

```
.content {
  padding: 16px 12px !important;
}

.card {
  margin-bottom: 16px !important;
  border-radius: 12px !important;
  box-shadow: 0 8px 32px rgba(0, 0, 0, 0.15) !important;
}

.card-header {
  padding: 16px 20px !important;
  flex-direction: column;
  gap: 12px;
  text-align: center;
}

.card-header.bg-dark.text-white {
  padding: 16px 20px !important;
}

.card-header h5,
.card-header h6 {
  font-size: 1rem;
}

#rangoSemana {
  margin: 8px 0 !important;
  padding: 12px 16px !important;
  font-size: 0.9rem !important;
  border-radius: 12px !important;
}

.p-4 {
  padding: 16px !important;
}

.p-3 {
  padding: 12px !important;
}

.p-2 {
  padding: 8px !important;
}

@media (max-width: 576px) {
  body::before,
```

```
body::after {
  display: none;
}

.content {
  padding: 12px 8px !important;
}

.card {
  margin-bottom: 12px !important;
  border-radius: 12px !important;
}

.card-header {
  padding: 12px 16px !important;
}

.card-header.bg-dark.text-white {
  padding: 12px 16px !important;
}

.card-header h5,
.card-header h6 {
  font-size: 0.95rem;
}

#rangoSemana {
  margin: 8px 0 !important;
  padding: 10px 16px !important;
  font-size: 0.85rem !important;
  text-align: center !important;
  border-radius: 12px !important;
}

.table-responsive {
  font-size: 0.8rem !important;
}

.table thead th {
  padding: 12px 8px !important;
  font-size: 0.75rem !important;
  white-space: nowrap;
}

.table tbody td {
  padding: 10px 8px !important;
  font-size: 0.8rem !important;
  white-space: nowrap;
```

```
}

.table .btn {
  padding: 4px 8px !important;
  font-size: 0.7rem !important;
}

.form-control,
.form-control-sm {
  font-size: 0.85rem !important;
  padding: 10px 12px !important;
  border-radius: 10px !important;
}

.form-label {
  font-size: 0.8rem !important;
}

.btn-sm {
  padding: 6px 12px !important;
  font-size: 0.8rem !important;
}

.alert-warning {
  padding: 12px 16px !important;
}

.alert-warning h6 {
  font-size: 0.85rem !important;
  margin-bottom: 8px !important;
}

.alert-warning ul {
  font-size: 0.8rem !important;
  padding-left: 20px !important;
}

.p-4 {
  padding: 12px !important;
}

.p-3 {
  padding: 8px !important;
}

.p-2 {
  padding: 6px !important;
}
```

```
}

@media (max-width: 480px) {
  .content {
    padding: 8px 4px !important;
  }

  .card {
    margin-bottom: 8px !important;
  }

  .card-header {
    padding: 10px 12px !important;
  }

  .alert {
    padding: 10px 12px !important;
    margin-bottom: 8px !important;
  }

  .table thead th {
    padding: 8px 4px !important;
    font-size: 0.7rem !important;
  }

  .table tbody td {
    padding: 8px 4px !important;
    font-size: 0.75rem !important;
  }

  .btn {
    padding: 6px 10px !important;
    font-size: 0.75rem !important;
  }

  #rangoSemana {
    font-size: 0.8rem !important;
    padding: 8px 12px !important;
  }
}

/* === TRANSICIONES GLOBALES === */
* {
  transition: all 0.3s ease;
}

/* === EFECTOS ESPECIALES === */
.card:hover .card-header i {
```

```

        transform: scale(1.2);
        text-shadow: 0 4px 12px rgba(249, 196, 31, 0.6);
    }

    .table tbody tr:hover .btn {
        transform: scale(1.05);
    }

    .tabla-ajustada td {
        white-space: normal;
        word-break: break-word;
    }

/* === RESPONSIVE FINAL SIN OCULTAR TH - Cumple.css === */
@media (max-width: 480px) {
    html, body {
        overflow-x: hidden !important;
    }

    .content,
    .container,
    .table-responsive,
    .card,
    .card-body {
        max-width: 100% !important;
        overflow-x: auto !important;
        box-sizing: border-box;
    }

    .table {
        table-layout: auto !important;
        min-width: 700px !important; /* Para evitar que se compriman columnas
    */
    }
}

.table thead th,
.table tbody td {
    white-space: nowrap !important;
    text-align: center !important;
    vertical-align: middle !important;
}

.table .btn {
    padding: 0.4rem 0.75rem !important;
    font-size: 0.75rem !important;
    min-width: 2rem !important;
    display: inline-flex !important;
    justify-content: center;
}

```

```

    align-items: center;
}

.table .badge {
  display: inline-block !important;
  min-width: 80px;
  text-align: center;
}
}

@media print {
  body * {
    visibility: hidden !important;
  }
}

#historialCumples, #historialCumples * {
  visibility: visible !important;
}

#historialCumples {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  background-color: white !important;
  padding: 20px;
}

.btn,
header,
nav,
footer,
aside {
  display: none !important;
}
}

/* === ALTURA FIJA Y SCROLL INVISIBLE PARA TABLAS === */

/* Div "Cumpleaños de la Semana" con altura fija y scroll */
.card-body .table-responsive {
  max-height: 400px !important; /* Ajusta esta altura según necesites */
  overflow-y: auto !important;
  overflow-x: auto !important; /* Scroll horizontal habilitado */
  scrollbar-width: none !important; /* Firefox - ocultar scrollbar */
  -ms-overflow-style: none !important; /* IE/Edge - ocultar scrollbar */
}

```

```
/* Ocultar scrollbar en WebKit (Chrome, Safari, Edge) - AMBOS scrollbars */
.card-body .table-responsive::-webkit-scrollbar {
    display: none !important;
}

.card-body .table-responsive::-webkit-scrollbar-horizontal {
    display: none !important;
}

.card-body .table-responsive::-webkit-scrollbar-vertical {
    display: none !important;
}

/* Específico para el historial que necesita más altura */
.card.shadow-sm .card-body .table-responsive {
    max-height: 500px !important; /* Altura mayor para el historial */
    overflow-y: auto !important;
    overflow-x: auto !important; /* Scroll horizontal habilitado */
    scrollbar-width: none !important;
    -ms-overflow-style: none !important;
}

.card.shadow-sm .card-body .table-responsive::-webkit-scrollbar {
    display: none !important;
}

.card.shadow-sm .card-body .table-responsive::-webkit-scrollbar-horizontal {
    display: none !important;
}

.card.shadow-sm .card-body .table-responsive::-webkit-scrollbar-vertical {
    display: none !important;
}

/* Asegurar que los contenedores internos no desborden */
#cumpleLista,
#historialCumples {
    width: 100% !important;
    box-sizing: border-box !important;
}

/* Smooth scroll para mejor experiencia */
.card-body .table-responsive {
    scroll-behavior: smooth !important;
}
```

```

/* Efecto fade en los bordes para indicar que hay más contenido */
.card-body .table-responsive::before {
    content: "";
    position: sticky;
    top: 0;
    left: 0;
    right: 0;
    height: 10px;
    background: linear-gradient(to bottom, rgba(255,255,255,1) 0%,
rgba(255,255,255,0) 100%);
    z-index: 10;
    pointer-events: none;
}

.card-body .table-responsive::after {
    content: "";
    position: sticky;
    bottom: 0;
    left: 0;
    right: 0;
    height: 10px;
    background: linear-gradient(to top, rgba(255,255,255,1) 0%,
rgba(255,255,255,0) 100%);
    z-index: 10;
    pointer-events: none;
}

/* Responsive - Ajustar alturas en dispositivos móviles */
@media (max-width: 768px) {
    .card-body .table-responsive {
        max-height: 300px !important;
    }

    .card.shadow-sm .card-body .table-responsive {
        max-height: 350px !important;
    }
}

@media (max-width: 576px) {
    .card-body .table-responsive {
        max-height: 250px !important;
    }

    .card.shadow-sm .card-body .table-responsive {
        max-height: 300px !important;
    }
}

```

```

/* === ESTILOS PARA BUSCADOR DE HISTORIAL DE CUMPLEAÑOS === */

/* Input de búsqueda por cédula */
#buscadorCedula {
    border-left: 3px solid #f9c41f !important;
    transition: all 0.3s ease !important;
}

#buscadorCedula:focus {
    border-left-color: #e6b619 !important;
    box-shadow: 0 0 0 0.2rem rgba(249, 196, 31, 0.25) !important;
}

/* Botón limpiar búsqueda */
#limpiarBusqueda {
    border-color: #f9c41f !important;
    color: #f9c41f !important;
    transition: all 0.3s ease !important;
}

#limpiarBusqueda:hover {
    background-color: #f9c41f !important;
    color: #000 !important;
    transform: scale(1.05);
}

/* Alertas pequeñas para mensajes de resultado */
.alert-sm {
    font-size: 0.875rem !important;
    border-radius: 8px !important;
}

/* Animación suave para filas de la tabla cuando se filtran */
#tablaHistorial tbody tr {
    transition: opacity 0.3s ease, transform 0.2s ease !important;
}

#tablaHistorial tbody tr[style*="display: none"] {
    opacity: 0;
    transform: translateX(-10px);
}

/* Efecto hover mejorado para el input group */
.input-group:hover #buscadorCedula {
    border-left-width: 4px !important;
    border-left-color: #e6b619 !important;
}

```

```

/* Mejorar el ícono de búsqueda */
.i.fa-search {
    transition: transform 0.3s ease;
}

.input-group:hover i.fa-search {
    transform: scale(1.1);
}

/* Efecto focus en el contenedor completo */
.input-group:focus-within {
    transform: translateY(-2px);
    box-shadow: 0 4px 12px rgba(249, 196, 31, 0.15);
}

/* Estilo para el texto de ayuda */
.text-muted i.fa-info-circle {
    color: #f9c41f !important;
    opacity: 0.7;
}

/* Responsive para el buscador */
@media (max-width: 576px) {
    #buscadorCedula {
        font-size: 0.875rem !important;
        padding: 8px 12px !important;
    }

    #limpiarBusqueda {
        padding: 8px 10px !important;
    }
}

/* === ANIMACIÓN === */
@keyframes bigPulse {
    0% {
        transform: scale(1);
    }
    50% {
        transform: scale(2);
        box-shadow: 0 0 50px rgba(249, 196, 31, 0.9);
    }
    100% {
        transform: scale(1);
    }
}

```

```

/* === BOTÓN ENVIAR CORREO === */
#btnEnviarCorreo {
    transition: all 0.3s ease;
}

#btnEnviarCorreo:focus {
    outline: 3px solid rgba(249, 196, 31, 0.4);
    outline-offset: 2px;
}

/* === CLASE DE INTERACCIÓN === */
.btn-highlight {
    animation: bigPulse 0.8s ease-in-out !important;
}

```

### | | └─ Dashboard.css

```

/*
Dashboard.css
-----
- Propósito: estilos para el panel principal (dashboard) con tarjetas
métricas, sección de tareas y layout responsivo.
- Estructura: variables en :root, layout y background, componentes
reutilizables (tarjetas, botones, listas) y media queries.
- Uso: mantener las clases/IDs en el HTML para que los estilos se
apliquen correctamente.
*/

:root {
/* Dashboard.css - versión moderna con fondo decorativo y sombras
avanzadas */
:root {
    --amarillo: #f9c41f;
    --negro: #121212;
    --blanco: #fff;
    --gris: #838886;
    --gris-fondo: #f7f7f7;
    --gris-borde: #e0e0e0;
    --sombra: rgba(0, 0, 0, 0.1);
    --sombra-hover: rgba(0, 0, 0, 0.15);
    --sombra-moderna: rgba(0, 0, 0, 0.06);
    --sombra-elevada: 0 8px 32px rgba(0, 0, 0, 0.12);
}

/* Layout base del dashboard */
/*
- 'display: flex' permite un sidebar fijo a la izquierda y área de
contenido flexible.

```

```

    - Las variables en :root controlan colores y sombras; cambiar allí para
theming.
*/



body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background: var(--gris) !important;
    margin: 0;
    color: var(--negro);
    display: flex;
    min-height: 100vh;
    overflow-x: hidden;
}

/* Fondo decorativo moderno */
/* Fondos decorativos (pseudo-elementos)
   - Añaden color y profundidad sin introducir elementos HTML adicionales.
   - pointer-events: none para no interferir con la interacción. */

body::before,
body::after {
    content: "";
    position: fixed;
    z-index: -1;
    pointer-events: none;
}

body::before {
    top: 0;
    left: 0;
    width: 40%;
    height: 70%;
    background: var(--amarillo);
}

body::after {
    bottom: 0;
    right: 0;
    width: 60%;
    height: 50%;
    background: var(--gris);
}

/* Contenedor principal del dashboard
   - margin-left: 220px asume un sidebar; ajustar si cambia la estructura
del layout. */

#container-dashboard {

```

```

margin-left: 220px;
padding: 24px;
flex: 1;
max-width: none;
margin-right: 0;
}

/* Título principal del dashboard */

.titulo-dashboard {
  font-size: 2rem;
  margin-bottom: 2rem;
  text-align: center;
  color: var(--blanco);
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);
}

/* Sección: resumen de métricas
 - Grid responsivo que adapta el número de columnas según el espacio
 disponible. */

.resumen-metricas {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));
  gap: 32px;
  margin-bottom: 40px;
}

/* Tarjeta métrica (tarjeta-metrica)
 - Reutilizable para mostrar icono, título, valor y enlace.
 - Pseudo-elemento ::before crea la línea decorativa superior. */

.tarjeta-metrica {
  background: rgba(255, 255, 255, 0.95);
  border-radius: 16px;
  box-shadow: var(--sombra-elevada);
  padding: 28px 24px;
  display: flex;
  flex-direction: column;
  align-items: flex-start;
  min-height: 190px;
  border: none;
  transition: all 0.3s ease;
  position: relative;
  overflow: hidden;
}

.tarjeta-metrica::before {

```

```
content: "";
position: absolute;
top: 0;
left: 0;
width: 100%;
height: 4px;
background: linear-gradient(90deg, var(--amarillo) 0%, #ffd700 100%);
}

.tarjeta-metrica:hover {
  transform: translateY(-8px);
  box-shadow: 0 12px 40px rgba(0, 0, 0, 0.18);
}

/* Icono de la métrica */

.icon-metrica {
  font-size: 2rem;
  color: var(--amarillo);
  margin-bottom: 8px;
  text-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

.tarjeta-metrica h4 {
  margin: 12px 0 8px 0;
  font-size: 1.15rem;
  font-weight: 600;
  color: var(--negro);
  line-height: 1.3;
}

.valor {
  font-size: 1.8rem;
  font-weight: bold;
  margin: 8px 0 6px 0;
  color: var(--negro);
  text-shadow: 0 1px 2px rgba(0, 0, 0, 0.1);
}

.subvalor {
  font-size: 1rem;
  color: #666;
  line-height: 1.4;
}

.link-metrica {
  margin-top: auto;
  color: var(--amarillo);
```

```

    font-size: 0.95rem;
    text-decoration: none;
    font-weight: 500;
    transition: all 0.3s ease;
    padding: 8px 0;
}

.link-metrica:hover {
    color: #e6b01f;
    text-decoration: underline;
    transform: translateX(4px);
}

/* Texto auxiliar para sección de cumpleaños */

#cumple-texto {
    font-size: 0.95em;
    color: #333;
}

/* Sección de tareas (todo)
 - Contiene formulario para añadir tareas y lista de tareas con estados.
 - Reutilizar .seccion-tareas para otras listas de acciones. */

.seccion-tareas {
    background: rgba(255, 255, 255, 0.95);
    border-radius: 16px;
    box-shadow: var(--sombra-elevada);
    padding: 32px 28px;
    margin-top: 40px;
    border: none;
    transition: all 0.3s ease;
    position: relative;
    overflow: hidden;
}

.seccion-tareas::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 4px;
    background: linear-gradient(90deg, var(--amarillo) 0%, #ffd700 100%);
}

.seccion-tareas:hover {
    box-shadow: 0 12px 40px rgba(0, 0, 0, 0.15);
}

```

```
}

/* Título de la lista de tareas */

.todo-title {
  font-size: 1.2rem;
  margin-bottom: 18px;
  display: flex;
  align-items: center;
  gap: 8px;
}

/* Formulario para añadir tareas
   - En escritorio usa layout en línea; en móvil se adapta vía media
queries. */

.formulario-tarea {
  display: flex;
  gap: 8px;
  margin-bottom: 16px;
}
.formulario-tarea input[type="text"] {
  flex: 1;
  padding: 12px 16px;
  border: 2px solid rgba(224, 224, 224, 0.6);
  border-radius: 8px;
  font-size: 1rem;
  transition: all 0.3s ease;
  background: rgba(255, 255, 255, 0.9);
}
.formulario-tarea input[type="text"]:focus {
  outline: none;
  border-color: var(--amarillo);
  box-shadow: 0 0 0 3px rgba(249, 196, 31, 0.2);
}
.formulario-tarea button {
  background: var(--amarillo);
  color: var(--negro);
  border: none;
  border-radius: 8px;
  padding: 12px 20px;
  font-size: 1rem;
  font-weight: 600;
  cursor: pointer;
  transition: all 0.3s ease;
  box-shadow: 0 4px 12px rgba(249, 196, 31, 0.3);
}
```

```
}

.formulario-tarea button:hover {
  background: #e6b01f;
  transform: translateY(-2px);
  box-shadow: 0 6px 20px rgba(249, 196, 31, 0.4);
}

/* Lista de tareas: cada item es una tarjeta compacta con acciones */

.todo-list {
  list-style: none;
  padding: 0;
  margin: 0;
}

/* Item de la lista de tareas */

.todo-list-item {
  background: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(224, 224, 224, 0.5);
  border-radius: 12px;
  padding: 18px 16px;
  margin-bottom: 16px;
  display: flex;
  justify-content: space-between;
  align-items: flex-start;
  transition: all 0.3s ease;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.05);
}

.todo-list-item:hover {
  transform: translateY(-2px);
  box-shadow: 0 6px 20px rgba(0, 0, 0, 0.1);
  background: rgba(255, 255, 255, 0.95);
}

/* Estado de tarea: completada (verde) */
/* Estado: tarea completada */

.estado-completada {
  color: #1a7f3c;
  font-weight: bold;
}

/* Estado de tarea: pendiente/incompleta (amarillo) */
/* Estado: tarea pendiente/incompleta */

.estado-pendiente {
```

```
color: #f9c41f;
font-weight: bold;
}

.contenido-tarea {
  font-size: 1rem;
  color: var(--negro);
  max-width: 70%;
  word-break: break-word;
}
.contador-caracteres {
  font-size: 0.85rem;
  color: #666;
  margin-top: 0.25rem;
}

.contador-peligro {
  color: #d63031; /* Rojo cuando quedan pocos caracteres */
  font-weight: bold;
}
.btn-cambiar, .btn-eliminar {
  border: none;
  border-radius: 8px;
  padding: 8px 16px;
  font-size: 0.9rem;
  font-weight: 500;
  cursor: pointer;
  margin-left: 8px;
  transition: all 0.3s ease;
  box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);
}
.btn-cambiar {
  background: var(--amarillo);
  color: var(--negro);
}
.btn-cambiar:hover {
  background: #e6b01f;
  transform: translateY(-1px);
  box-shadow: 0 4px 12px rgba(249, 196, 31, 0.3);
}
.btn-eliminar {
  background: #e74c3c;
  color: #fff;
}
```

```
.btn-eliminar:hover {
  background: #c0392b;
  transform: translateY(-1px);
  box-shadow: 0 4px 12px rgba(231, 76, 60, 0.3);
}

.info-tarea {
  margin-top: 0.5rem;
  display: flex;
  justify-content: space-between;
  align-items: center;
  font-size: 0.9rem;
}

.fecha-tarea {
  color: #888;
  font-size: 0.85rem;
  white-space: nowrap;
}

@media (max-width: 992px) {
  #container-dashboard {
    margin-left: 0;
    padding: 16px;
  }

  .resumen-metricas {
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
    gap: 24px;
  }
}

@media (max-width: 700px) {
  #container-dashboard {
    padding: 12px 8px;
  }

  .resumen-metricas {
    gap: 16px;
    grid-template-columns: 1fr;
  }
}

.tarjeta-metrica {
  padding: 20px 16px;
  min-height: 160px;
}

.seccion-tareas {
  padding: 20px 16px;
```

```

}

.todo-list-item {
  flex-direction: column;
  align-items: stretch;
  gap: 12px;
}

.btn-cambiar, .btn-eliminar {
  width: 100%;
  margin-left: 0;
  margin-bottom: 8px;
}

.btn-eliminar {
  margin-bottom: 0;
}

.formulario-tarea {
  flex-direction: column;
  gap: 12px;
}
}

@media (max-width: 480px) {
  body::before,
  body::after {
    display: none;
  }

  body {
    background: var(--gris-fondo) !important;
  }
}
}

```

## | | └─ Layout.css

```

/* Layout.css – documentación in-situ
*/

html {
  scrollbar-width: none; /* Firefox */
  -ms-overflow-style: none; /* IE y Edge */
}
html::-webkit-scrollbar {
  display: none; /* Chrome, Safari, Opera */
}

```

```

/* Scroll invisible pero funcional en todas las tablas con scroll de
todas las secciones */
.table-responsive {
    scrollbar-width: none; /* Firefox */
    -ms-overflow-style: none; /* IE y Edge */
}
.table-responsive::-webkit-scrollbar {
    display: none; /* Chrome, Safari, Opera */
}
/* Layout.css - versión moderna con sidebar elegante y navegación fluida
*/
/* === Variables globales ===
   Colores y valores reutilizables para mantener consistencia visual.
   Cambiar aquí para modificar temas (amarillo institucional, sombras,
anchos).
*/
:root {
    --amarillo: #f9c41f;
    --gris: #838886;
    --negro: #000000;
    --blanco: #ffffff;
    --gris-fondo: #f7f7f7;
    --gris-borde: #e0e0e0;
    --sombra: rgba(0, 0, 0, 0.1);
    --sombra-hover: rgba(0, 0, 0, 0.15);
    --sombra-moderna: rgba(0, 0, 0, 0.06);
    --sombra-elevada: 0 8px 32px rgba(0, 0, 0, 0.12);
    --sidebar-width: 220px;
}

body {
    margin: 0;
    font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
    background: var(--gris) !important;
    display: flex;
    min-height: 100vh;
    overflow-x: hidden;
    /* Ocultar scrollbars en todos los navegadores */
    scrollbar-width: none; /* Firefox */
    -ms-overflow-style: none; /* IE y Edge */
}

body::-webkit-scrollbar {
    display: none; /* Chrome, Safari y Opera */
}

/* Fondo decorativo moderno */
body::before,

```

```

body::after {
    content: "";
    position: fixed;
    z-index: -1;
    pointer-events: none;
}

body::before {
    top: 0;
    left: 0;
    width: 40%;
    height: 70%;
    background: var(--amarillo);
}

body::after {
    bottom: 0;
    right: 0;
    width: 60%;
    height: 50%;
    background: var(--gris);
}

/* Bloquear scroll horizontal */
html, body {
    max-width: 100vw;
    overflow-x: hidden;
}

/* SIDEBAR MODERNO */
/* === SIDEBAR (menú lateral) ===
   Estilos para la barra lateral fija: fondo oscuro, ancho controlado
por
   la variable --sidebar-width y comportamiento responsive. La clase
   `activa` se usa en móvil para mostrar el sidebar. Mantener cuidado
con
   overflow y scroll invisible para que la experiencia sea limpia.
*/
.sidebar {
    width: var(--sidebar-width);
    background: linear-gradient(135deg, var(--negro) 0%, #1a1a1a 100%);
    color: var(--blanco);
    padding: 24px 20px;
    position: fixed;
    top: 0;
    bottom: 0;
    left: 0;
    overflow-y: auto;
}

```

```
    box-shadow: var(--sombra-elevada);
    z-index: 1000;
    transition: all 0.3s ease;
}

/* Sidebar: solo el enlace activo es amarillo */
.sidebar ul li a.active {
    background: var(--amarillo) !important;
    color: var(--negro) !important;
    box-shadow: 0 4px 16px rgba(249, 196, 31, 0.15);
    border-radius: 12px;
    font-weight: 700;
}

.sidebar ul li a {
    background: transparent;
    color: var(--blanco);
    border-radius: 12px;
    font-weight: 500;
    transition: background 0.2s, color 0.2s;
}

.sidebar ul li a:not(.active):hover {
    background: rgba(249, 196, 31, 0.08);
    color: var(--amarillo);
}

/* Hacer scroll invisible en el sidebar */
.sidebar {
    scrollbar-width: none; /* Firefox */
    -ms-overflow-style: none; /* IE y Edge */
}

.sidebar::-webkit-scrollbar {
    display: none; /* Chrome, Safari y Opera */
}

/* HEADER DEL SIDEBAR MODERNIZADO */
.sidebar-header {
    display: flex;
    align-items: center;
    gap: 12px;
    margin-bottom: 2rem;
    padding: 16px 12px;
    border-radius: 12px;
    background: rgba(249, 196, 31, 0.1);
    border: 1px solid rgba(249, 196, 31, 0.2);
    position: relative;
```

```
        overflow: hidden;
    }

.btn-cumple-mini {
    position: absolute;
    top: 6px;           /* ajusta fino según tu header */
    right: 8px;          /* posición donde marcaste en rojo */
    display: inline-flex;
    align-items: center;
    justify-content: center;
    width: 26px;
    height: 26px;
    border-radius: 50%;
    border: 1px solid #000;
    background: #F9C41F;      /* amarillo institucional */
    cursor: pointer;
    box-shadow: 0 2px 6px rgba(0,0,0,.3);
    transition: transform .1s ease, box-shadow .15s ease, background-color
    .2s;
}

.btn-cumple-mini i {
    font-size: 14px;
    line-height: 1;
}

.btn-cumple-mini:hover {
    transform: translateY(-1px);
    box-shadow: 0 4px 10px rgba(0,0,0,.35);
}

.btn-cumple-mini:active {
    transform: translateY(0);
    box-shadow: 0 2px 6px rgba(0,0,0,.3);
}

.sidebar-header::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 2px;
    background: linear-gradient(90deg, var(--amarillo) 0%, #ffd700 100%);
}

.img-header {
    width: 44px;
```

```
height: 44px;
object-fit: contain;
border-radius: 8px;
background: rgba(249, 196, 31, 0.2);
padding: 6px;
transition: all 0.3s ease;
box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
}

.img-header:hover {
    transform: scale(1.1) rotate(5deg);
    box-shadow: 0 4px 16px rgba(249, 196, 31, 0.3);
}

.sidebar-header h3 {
    font-size: 1.25rem;
    margin: 0;
    font-weight: 700;
    color: var(--amarillo);
    text-shadow: 0 1px 2px rgba(0, 0, 0, 0.3);
}

/* MENÚ DEL SIDEBAR MODERNIZADO */
.sidebar ul {
    list-style: none;
    padding: 0;
    margin: 0;
}

.sidebar ul li {
    margin-bottom: 8px;
}

.sidebar ul li a {
    color: var(--blanco);
    text-decoration: none;
    padding: 14px 16px;
    display: flex;
    align-items: center;
    gap: 12px;
    border-radius: 12px;
    transition: all 0.3s ease;
    font-size: 0.95rem;
    font-weight: 500;
    position: relative;
    overflow: hidden;
}
```

```
.sidebar ul li a::before {
  content: "";
  position: absolute;
  left: 0;
  top: 0;
  width: 0;
  height: 100%;
  background: linear-gradient(90deg, var(--amarillo) 0%, #ffd700 100%);
  transition: width 0.3s ease;
  z-index: -1;
}

.sidebar ul li a i {
  font-size: 1.1rem;
  width: 20px;
  text-align: center;
  transition: all 0.3s ease;
}

.sidebar ul li a:hover {
  background: rgba(249, 196, 31, 0.15);
  color: var(--amarillo);
  transform: translateX(4px);
  box-shadow: 0 4px 12px rgba(249, 196, 31, 0.2);
}

.sidebar ul li a:hover::before {
  width: 4px;
}

.sidebar ul li a:hover i {
  transform: scale(1.1);
  color: var(--amarillo);
}

.sidebar ul li a.active {
  background: linear-gradient(135deg, var(--amarillo) 0%, #ffd700
100%);
  color: var(--negro);
  font-weight: 600;
  box-shadow: 0 6px 20px rgba(249, 196, 31, 0.4);
  transform: translateX(2px);
}

.sidebar ul li a.active i {
  color: var(--negro);
  transform: scale(1.1);
}
```

```

/* CONTENIDO PRINCIPAL MODERNIZADO */
/* === CONTENIDO PRINCIPAL y TOPBAR ===
   .content contiene la vista principal desplazada por el ancho del
sidebar.
   .topbar es un encabezado con sombra que sirve para botones rápidos y
título.
   Evitar usar !important a menos que sea estrictamente necesario.
*/
.content {
  margin-left: var(--sidebar-width);
  flex-grow: 1;
  padding: 24px;
  min-height: 100vh;
}

.topbar {
  background: rgba(255, 255, 255, 0.95);
  padding: 20px 28px;
  border-radius: 16px;
  box-shadow: var(--sombra-elevada);
  margin-bottom: 24px;
  border: none;
  transition: all 0.3s ease;
  position: relative;
  overflow: hidden;
}

.topbar::before {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 3px;
  background: linear-gradient(90deg, var(--amarillo) 0%, #ffd700 100%);
}

.topbar:hover {
  box-shadow: 0 12px 40px rgba(0, 0, 0, 0.15);
  transform: translateY(-2px);
}

.topbar h4 {
  margin: 0;
  color: var(--negro);
  font-weight: 600;
  font-size: 1.25rem;
}

```

```

}

/* ===== */
/* ELEMENTOS DE ESTADO MODERNIZADOS */
/* ===== */

/* Links deshabilitados */
/* === Estados y utilidades accesibilidad ===
   Clases de estado (deshabilitado / solo lectura / acceso restringido)
que
   se reutilizan en diferentes vistas para indicar elementos inactivos.
*/
.disabled-link {
    color: rgba(255, 255, 255, 0.4) !important;
    cursor: not-allowed !important;
    background: rgba(255, 255, 255, 0.05) !important;
    opacity: 0.6 !important;
    pointer-events: none !important;
    border-radius: 8px !important;
}

.disabled-link i {
    color: rgba(255, 255, 255, 0.4) !important;
}

/* Botones deshabilitados */
.btn-disabled {
    opacity: 0.6 !important;
    cursor: not-allowed !important;
    filter: grayscale(50%);
    transition: all 0.3s ease;
}

/* Campos de formulario de solo lectura */
.form-readonly {
    background: rgba(248, 249, 250, 0.8) !important;
    color: #6c757d !important;
    cursor: not-allowed !important;
    border: 2px solid rgba(224, 224, 224, 0.5) !important;
    border-radius: 8px !important;
}

/* Elementos con acceso restringido */
.access-restricted {
    opacity: 0.5 !important;
    pointer-events: none !important;
    cursor: not-allowed !important;
    filter: grayscale(30%);
```

```

}

/* ===== */
/* MENÚ HAMBURGUESA MODERNO */
/* ===== */
/* === Botón hamburguesa (móvil) ===
   .btn-menu-toggle aparece en tamaños pequeños. Se recomienda controlar
   su visibilidad desde JS y alternar la clase .activa en .sidebar.
*/
.btn-menu-toggle {
  display: none;
  position: fixed;
  top: 20px;
  left: 20px;
  z-index: 1001;
  background: linear-gradient(135deg, var(--amarillo) 0%, #ffd700
100%);
  border: none;
  border-radius: 12px;
  padding: 12px 16px;
  font-size: 1.4rem;
  color: var(--negro);
  cursor: pointer;
  box-shadow: 0 6px 20px rgba(249, 196, 31, 0.4);
  transition: all 0.3s ease;
}

.btn-menu-toggle:hover {
  transform: translateY(-2px) scale(1.05);
  box-shadow: 0 8px 25px rgba(249, 196, 31, 0.5);
}

.btn-menu-toggle:active {
  transform: translateY(0) scale(0.95);
}

/* RESPONSIVIDAD MODERNA */
/* === Breakpoints importantes ===
   Ajustes principales para tabletas y móviles. Revisar estos media
queries
   cuando se cambie el layout general o el ancho del sidebar.
*/
@media (max-width: 992px) {
  .content {
    margin-left: 0;
    padding: 20px 16px;
  }
}

```

```

.topbar {
    padding: 16px 20px;
    margin-bottom: 20px;
}
}

@media (max-width: 768px) {
    .btn-menu-toggle {
        display: block;
    }
}

.sidebar {
    transform: translateX(-100%);
    transition: transform 0.4s cubic-bezier(0.4, 0, 0.2, 1);
    z-index: 1000;
    box-shadow: 0 0 50px rgba(0, 0, 0, 0.3);
}

.sidebar.activa {
    transform: translateX(0);
}

.content {
    margin-left: 0;
    padding: 16px 12px;
    padding-top: 80px; /* espacio para el botón hamburguesa */
}

.topbar {
    padding: 16px 20px;
    border-radius: 12px;
}

/* Overlay cuando el sidebar está abierto */
.sidebar.activa::after {
    content: "";
    position: fixed;
    top: 0;
    left: 100%;
    width: 100vw;
    height: 100vh;
    background: rgba(0, 0, 0, 0.5);
    z-index: -1;
}
}

@media (max-width: 480px) {
    body::before,

```

```
body::after {
    display: none;
}

body {
    background: var(--gris-fondo) !important;
}

.content {
    padding: 12px 8px;
    padding-top: 75px;
}

.topbar {
    padding: 12px 16px;
    margin-bottom: 16px;
}

.sidebar {
    width: 280px;
    padding: 20px 16px;
}
}

@media (max-width: 320px) {
    html,
    body {
        max-width: 100vw;
        overflow-x: hidden;
    }
}

.sidebar {
    width: 100vw;
    padding: 16px 12px;
}

.sidebar ul li a {
    font-size: 0.9rem;
    padding: 12px 14px;
}

.img-header {
    width: 36px;
    height: 36px;
}

.sidebar-header h3 {
    font-size: 1.1rem;
```

```
}

.topbar h4 {
    font-size: 1rem;
    text-align: center;
}

.content {
    padding: 10px 6px;
    padding-top: 70px;
}

.btn-menu-toggle {
    padding: 10px 14px;
    font-size: 1.2rem;
}
}

/* ===== */
/* NOTA: estilos obsoletos comentados más abajo por compatibilidad. */
/* ===== */

/* === Sidebar sticky header === */
.sidebar {
    display: flex;
    flex-direction: column;
    height: 100vh;
    /* el resto de tu sidebar ya lo tiene */
}

.sidebar-header {
    position: sticky;
    top: 0;
    z-index: 2;
    background: rgba(249,196,31,0.10);
    /* Repite border-radius y border si lo deseas, o déjalo así */
    /* La sombra y padding ya la tienes */
}

/* El ul ocupa el resto del sidebar y hace scroll */
.sidebar ul {
    flex: 1 1 auto;
    min-height: 0;
    overflow-y: auto;
    /* Para evitar que quede muy pegado abajo: */
    padding-bottom: 1.5rem;
    /* Para evitar que el header lo tape al scrollear hacia arriba */
    scroll-padding-top: 88px; /* aprox header+gap */
}
```

```

}

/* RESPONSIVO: sidebar sigue sticky en móvil */
@media (max-width: 768px) {
    .sidebar {
        height: 100vh; /* mantiene altura fija en móvil también */
    }
    .sidebar-header {
        top: 0;
    }
    .sidebar ul {
        padding-bottom: 2rem;
    }
}

/* Scroll invisible para el menú de la sidebar */
.sidebar ul {
    scrollbar-width: none;           /* Firefox */
    -ms-overflow-style: none;        /* IE y Edge */
}

.sidebar ul::-webkit-scrollbar {
    display: none;                  /* Chrome, Safari, Opera */
}

/* ===== */
/* MODAL CUMPLEAÑOS
   Modal específico para mostrar la lista de cumpleaños. Se controla su
   visibilidad desde JS (cambiar display a 'flex' cuando se abra).
Incluye
   animación modalSlideIn y estilos para el cuerpo con scroll interno.
*/
/* ===== */

/* Animación para el modal */
@keyframes modalSlideIn {
    from {
        transform: translateY(-50px) scale(0.9);
        opacity: 0;
    }
    to {
        transform: translateY(0) scale(1);
        opacity: 1;
    }
}

.modal-cumples {
    position: fixed;

```

```
    top: 0;
    left: 0;
    width: 100vw;
    height: 100vh;
    margin: 0;
    padding: 0;
    background: rgba(0, 0, 0, 0.7);
    display: none;
    justify-content: center;
    align-items: center;
    z-index: 999999;
    backdrop-filter: blur(4px);
}

.modal-contenido-cumples {
    background: var(--blanco);
    color: var(--negro);
    border-radius: 16px;
    width: 90%;
    max-width: 800px;
    max-height: 85vh;
    box-shadow: var(--sombra-elevada);
    position: relative;
    overflow: hidden;
    animation: modalSlideIn 0.3s ease-out;
    display: flex;
    flex-direction: column;
}

.modal-header-cumples {
    background: linear-gradient(135deg, var(--amarillo), #ffd700);
    color: var(--negro);
    padding: 1.5rem 2rem;
    display: flex;
    justify-content: between;
    align-items: center;
    border-bottom: 1px solid var(--gris-borde);
}

.modal-header-cumples h3 {
    margin: 0;
    font-size: 1.5rem;
    font-weight: 600;
    flex: 1;
}

.btn-cerrar-cumples {
    background: none;
```

```

border: none;
font-size: 2rem;
font-weight: bold;
color: var(--negro);
cursor: pointer;
padding: 0;
width: 32px;
height: 32px;
display: flex;
align-items: center;
justify-content: center;
border-radius: 50%;
transition: all 0.3s ease;
}

.btn-cerrar-cumples:hover {
    background: rgba(0, 0, 0, 0.1);
    transform: scale(1.1);
}

.modal-body-cumples {
    flex: 1;
    padding: 1.5rem;
    overflow-y: auto;
    min-height: 0;
    /* Ocultar barra de scroll */
    scrollbar-width: none;          /* Firefox */
    -ms-overflow-style: none;       /* IE y Edge */
}

.modal-body-cumples::-webkit-scrollbar {
    display: none;                  /* Chrome, Safari, Opera */
}

.modal-footer-cumples {
    background: var(--gris-fondo);
    padding: 1rem 2rem;
    border-top: 1px solid var(--gris-borde);
    display: flex;
    justify-content: center;
}

.btn-ir-cumples {
    background: var(--amarillo);
    color: var(--negro);
    border: none;
    padding: 0.8rem 2rem;
    border-radius: 8px;
}

```

```

    font-size: 1rem;
    font-weight: 600;
    cursor: pointer;
    transition: all 0.3s ease;
    display: flex;
    align-items: center;
    gap: 0.5rem;
    box-shadow: 0 2px 8px rgba(249, 196, 31, 0.3);
}

.btn-ir-cumpleaños:hover {
    background: #e6b01f;
    transform: translateY(-2px);
    box-shadow: 0 4px 16px rgba(249, 196, 31, 0.4);
}

.cumpleaños-container {
    max-height: 400px;
    overflow-y: auto;
    /* Ocultar barra de scroll */
    scrollbar-width: none;          /* Firefox */
    -ms-overflow-style: none;        /* IE y Edge */
}

.cumpleaños-container::-webkit-scrollbar {
    display: none;                  /* Chrome, Safari, Opera */
}

.cumpleaño-card {
    background: var(--blanco);
    border: 1px solid var(--gris-borde);
    border-radius: 12px;
    padding: 1rem;
    margin-bottom: 1rem;
    box-shadow: var(--sombra-moderna);
    transition: all 0.3s ease;
    cursor: pointer;
}

.cumpleaño-card:hover {
    box-shadow: var(--sombra-hover);
    transform: translateY(-2px);
    border-color: var(--amarillo);
}

.cumpleaño-card.sin-correo {
    border-left: 4px solid #ff6b6b;
    background: rgba(255, 107, 107, 0.05);
}

```

```
}

.cumple-header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 0.5rem;
}

.cumple-nombre {
  font-size: 1.1rem;
  font-weight: 600;
  color: var(--negro);
  margin: 0;
}

.cumple-fecha {
  background: var(--amarillo);
  color: var(--negro);
  padding: 0.3rem 0.8rem;
  border-radius: 20px;
  font-size: 0.8rem;
  font-weight: 600;
}

.cumple-info {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
  gap: 0.5rem;
}

.cumple-detalles {
  font-size: 0.9rem;
  color: var(--gris);
}

.cumple-label {
  font-weight: 600;
  color: var(--negro);
}

.alerta-sin-cumples {
  text-align: center;
  padding: 3rem 2rem;
  color: var(--gris);
}

.alerta-sin-cumples i {
```

```

        font-size: 4rem;
        margin-bottom: 1rem;
        opacity: 0.5;
    }

    @media (max-width: 768px) {
        .modal-contenido-cumples {
            width: 95%;
            max-height: 90vh;
        }
    }

    .modal-header-cumples {
        padding: 1rem;
    }

    .modal-body-cumples {
        padding: 1rem;
    }

    .modal-footer-cumples {
        padding: 1rem;
    }

    .cumple-info {
        grid-template-columns: 1fr;
    }

    .btn-ir-cumples {
        width: 100%;
        justify-content: center;
    }
}

.sidebar ul li a:focus-visible,
.sidebar ul li a[aria-current="page"] {
    background: linear-gradient(135deg, var(--amarillo) 0%, #ffd700 100%);
    color: var(--negro);
    font-weight: 600;
    box-shadow: 0 6px 20px rgba(249, 196, 31, 0.4);
    transform: translateX(2px);
    outline: none;
}

/* ===== */
/* MODAL TARJETA (buscar por número)
   Modal reutilizable para buscar tarjetas por número. Se centra en
   pantalla
   y usa el mismo keyframe `modalSlideIn` para entrada. Mostrar desde
   JS.

```

```

*/
/* ===== */

.modal-tarjeta {
  position: fixed;
  inset: 0; /* top:0; right:0; bottom:0; left:0 */
  width: 100vw;
  height: 100vh;
  display: none; /* se muestra como flex desde JS */
  align-items: center;
  justify-content: center;
  background: rgba(0,0,0,0.7);
  backdrop-filter: blur(4px);
  z-index: 1000000; /* por encima del resto */
}

.modal-tarjeta .modal-contenido {
  width: min(500px, 92vw);
  background: var(--blanco);
  color: var(--negro);
  border-radius: 16px;
  box-shadow: var(--sombra-elevada);
  padding: 24px 20px;
  animation: modalSlideIn .25s ease-out; /* reutiliza el keyframes ya
definido */
}

.modal-tarjeta h3 {
  margin: 0 0 12px;
  font-size: 1.25rem;
  font-weight: 700;
}

.modal-tarjeta input {
  width: 100%;
  padding: 12px 14px;
  border: 2px solid var(--gris-borde);
  border-radius: 10px;
  outline: none;
  font-size: 1rem;
  transition: border .2s ease, box-shadow .2s ease;
}

.modal-tarjeta input:focus {
  border-color: var(--amarillo);
  box-shadow: 0 0 0 4px rgba(249,196,31,.2);
}

```

```
.modal-mensaje-error {
  color: #dc3545;
  min-height: 1.2em;
  margin-top: 8px;
  font-weight: 600;
}

.modal-botones {
  margin-top: 16px;
  display: flex;
  gap: 12px;
  justify-content: flex-end;
}

.modal-botones button {
  border: none;
  border-radius: 10px;
  padding: 10px 16px;
  font-weight: 600;
  cursor: pointer;
  box-shadow: 0 2px 8px rgba(0,0,0,.08);
  transition: transform .15s, box-shadow .15s;
}

.modal-botones button:first-child {
  background: var(--gris-fondo);
  color: var(--negro);
}

.modal-botones button:first-child:hover {
  transform: translateY(-1px);
  box-shadow: 0 4px 12px rgba(0,0,0,.12);
}

.modal-botones button:last-child {
  background: var(--amarillo);
  color: var(--negro);
}

.modal-botones button:last-child:hover {
  transform: translateY(-1px);
  box-shadow: 0 4px 14px rgba(249,196,31,.45);
}

@media (max-width: 480px){
  .modal-tarjeta .modal-contenido { padding: 18px 16px; }
}
```

```

/* Evita scroll del documento mientras un modal está abierto */
body.modal-open {
    overflow: hidden;
}

/* Redundante pero seguro: centra ambos tipos de modal */
.modal-tarjeta,
.modal-cumples {
    display: none;           /* se activa a 'flex' desde JS */
    align-items: center;
    justify-content: center;
    position: fixed;
    inset: 0;
}

```

### | |    Login.css

```

/* Login.css – documentación in-situ
   Propósito: estilos del formulario de inicio de sesión. Diseño moderno,
   centrado y responsive con énfasis en estado, accesibilidad y
consistencia
   visual.
*/
:root {
    --amarillo: #f4b400;
    --dorado: #ffd700;
    --dorado-claro: #ffed4e;
    --negro: #000000;
    --negro-suave: #1a1a1a;
    --blanco: #ffffff;
    --gris: #2a2a2a;
    --gris-claro: #e8eaed;
    --gris-muy-claro: #f8f9fa;
    --gris-oscuro: #4a4a4a;
    --gris-fondo: #f7f8fc;

    /* ⚡ Efectos mágicos */
    --gradiente-dorado: linear-gradient(135deg, var(--amarillo) 0%, var(--dorado) 50%, var(--dorado-claro) 100%);
    --gradiente-premium: linear-gradient(135deg, rgba(244, 180, 0, 0.15) 0%, rgba(255, 215, 0, 0.1) 50%, rgba(255, 237, 78, 0.05) 100%);
    --sombra-magica: 0 20px 60px rgba(244, 180, 0, 0.2);
    --sombra-elevada: 0 16px 48px rgba(0, 0, 0, 0.15);
    --resplandor-dorado: 0 0 40px rgba(244, 180, 0, 0.4);
}

/* -----
   Reset y configuración base
   Reglas globales que normalizan box-sizing y márgenes.

```

```

    Mantener al inicio para evitar surprises en layouts.

*/
/* 🖥 Reset y configuración base */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

/* -----
   Fondo y layout del body
   Aquí se define el fondo principal (gradiente) y el centrado absoluto
   del contenedor de login. En móviles se fuerza fondo blanco y se ocultan
   las partículas para mejorar rendimiento.
*/
/* 📱 FONDO UNIVERSAL - CENTRADO ABSOLUTO */
body {
    font-family: 'Segoe UI', -apple-system, BlinkMacSystemFont, 'Roboto',
    sans-serif;
    background: linear-gradient(135deg,
        #ffffff 0%,
        #f8f9fa 25%,
        #e9ecf 50%,
        #f8f9fa 75%,
        #ffffff 100%
    );
    /* Fallback sólido para móviles */
    background-color: #ffffff;
    min-height: 100vh;
    height: 100vh;
    /* Permitir que el contenido se ajuste siempre al centro sin scroll */
    overflow-y: auto;
    display: flex;
    align-items: center;
    justify-content: center;
    padding: clamp(8px, 1vw, 16px);
    overflow: hidden;
    position: relative;
    /* Centrado universal para cualquier tamaño */
    box-sizing: border-box;

    /* 📱 FORZAR FONDO BLANCO EN MÓVILES */
    background-attachment: fixed;
    -webkit-background-size: cover;
    background-size: cover;
    background-repeat: no-repeat;
}

```

```

/* ♦ PARTÍCULAS FLOTANTES */
body::before,
body::after {
    content: "";
    position: fixed;
    z-index: -1;
    pointer-events: none;
    border-radius: 50%;
}

body::before {
    top: -20%;
    left: -20%;
    width: 80%;
    height: 100%;
    background: radial-gradient(circle at 30% 40%,
        rgba(244, 180, 0, 0.08) 0%,
        rgba(255, 215, 0, 0.04) 30%,
        transparent 70%
    );
    animation: orbita-magica 20s linear infinite;
}

body::after {
    bottom: -25%;
    right: -25%;
    width: 90%;
    height: 110%;
    background: radial-gradient(circle at 70% 60%,
        rgba(108, 117, 125, 0.05) 0%,
        rgba(244, 180, 0, 0.03) 40%,
        transparent 80%
    );
    animation: orbita-magica 25s linear infinite reverse;
}

@keyframes orbita-magica {
    0% {
        transform: rotate(0deg) translateX(50px) rotate(0deg) scale(1);
        opacity: 0.3;
    }
    50% {
        transform: rotate(180deg) translateX(50px) rotate(-180deg) scale(1);
        opacity: 0.4;
    }
    100% {
        transform: rotate(360deg) translateX(50px) rotate(-360deg) scale(1);
        opacity: 0.3;
    }
}

```

```

    }

/*
----- Contenedor principal ` .login-container` -----
Contenedor centrado y con ancho máximo. Aquí puedes controlar el tamaño
máximo del formulario sin tocar la tarjeta interna.
*/
/* ✎ CONTENEDOR MÁS ESTRECHO - ELEGANTE Y CENTRADO */
.login-container {
  width: 100%;
  max-width: clamp(320px, 70vw, 380px);
  margin: 0 auto;
  position: relative;
  z-index: 10;
  animation: aparicion-epica 1.5s ease-out;
  /* Centrado matemático universal */
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  box-sizing: border-box;
  /* Límites elegantes sin rigidez - SOLO OPTIMIZACIÓN MÍNIMA */
  /* Eliminar max-height para que nunca corte el contenido */
  max-height: none;
  overflow: visible;
}

@keyframes aparicion-epica {
  0% {
    opacity: 0;
    transform: translateY(50px) scale(0.9);
  }
  100% {
    opacity: 1;
    transform: translateY(0) scale(1);
  }
}

/*
----- Tarjeta ` .login-card` -----
Estilos principales de la tarjeta: borde, sombra y máscara dorada.
Evitar usar `overflow:hidden` que pueda ocultar botones o errores.
*/
/* ✎ TARJETA MÁS ESTRECHA - PROPORCIÓN PERFECTA */
.login-card {
  background: var(--blanco);
  padding: clamp(2rem, 4.5vh, 3rem) clamp(1.5rem, 3.5vw, 2rem);
}

```

```

border-radius: clamp(1.5rem, 3vh, 2.2rem);
border: 2px solid transparent;
box-shadow: var(--sombra-elevada), var(--sombra-magica);
transition: all 0.4s cubic-bezier(0.25, 0.46, 0.45, 0.94);
position: relative;
overflow: hidden;
backdrop-filter: blur(25px);
width: 100%;
max-width: 100%;
/* Eliminar max-height y overflow para que el botón nunca quede oculto */
*/
max-height: none;
box-sizing: border-box;
overflow: visible;
scrollbar-width: none;
-ms-overflow-style: none;
}

.login-card::-webkit-scrollbar {
  display: none;
}

.login-card::before {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  border-radius: clamp(1.5rem, 3vh, 2.2rem);
  padding: 2px;
  background: var(--gradiente-dorado);

  /* ☑ SOLUCIÓN UNIVERSAL PARA TODOS LOS NAVEGADORES */
  /* Webkit (Chrome, Safari, Edge) */
  mask: linear-gradient(#fff 0 0) content-box, linear-gradient(#fff 0 0);
  -webkit-mask: linear-gradient(#fff 0 0) content-box, linear-
gradient(#fff 0 0);
  mask-composite: xor;
  -webkit-mask-composite: xor;

  /* 🐻 FIREFOX ESPECÍFICO - BORDE DORADO */
  border: 2px solid transparent;
  background-clip: padding-box;

  z-index: -1;
  opacity: 0.8;
}

```

```

/* 🦵 SOLUCIÓN ESPECÍFICA PARA FIREFOX */
@-moz-document url-prefix() {
  .login-card {
    border: 2px solid var(--amarillo) !important;
    background: var(--blanco) !important;
  }

  .login-card::before {
    display: none !important;
  }
}

/* 💬 DETECCIÓN MODERNA DE SOPORTE PARA MÁSCARA */
@supports not (mask-composite: xor) {
  .login-card {
    border: 2px solid var(--amarillo);
    background: var(--blanco);
  }

  .login-card::before {
    display: none;
  }
}

.login-card:hover {
  transform: translateY(-12px) scale(1.02);
  box-shadow:
    var(--sombra-elevada),
    var(--sombra-magica),
    var(--resplandor-dorado);
  border-color: rgba(244, 180, 0, 0.3);
}

/* -----
   Logo y elementos visuales
   `logo-container` , `logo` y `logo-bastos` definen tamaños y
animación.
   Mantener proporciones por responsive para evitar distorsión.
*/
/* 📺 LOGO ELEGANTE */
.logo-container {
  text-align: center;
  margin-bottom: clamp(0.8rem, 2vh, 1.2rem);
  position: relative;
  z-index: 2;
}

```

```

/* 🌟 LOGO BASTOS - COMPACTO Y ELEGANTE */
.logo-bastos {
    width: clamp(40px, 10vw, 55px);
    height: clamp(40px, 10vw, 55px);
    transition: all 0.4s ease;
    filter: drop-shadow(0 3px 12px rgba(244, 180, 0, 0.2));
    animation: bastos-resplendor 3s ease-in-out infinite;
    display: block;
    margin: 0 auto clamp(0.3rem, 0.8vh, 0.5rem) auto;
    opacity: 0.95;
}

@keyframes bastos-resplendor {
    0%, 100% {
        filter: drop-shadow(0 3px 12px rgba(244, 180, 0, 0.2));
        transform: scale(1);
        opacity: 0.95;
    }
    50% {
        filter: drop-shadow(0 4px 16px rgba(244, 180, 0, 0.3));
        transform: scale(1.02);
        opacity: 1;
    }
}

.logo-bastos:hover {
    transform: scale(1.05);
    filter: drop-shadow(0 6px 20px rgba(244, 180, 0, 0.4));
    opacity: 1;
}

.logo {
    width: clamp(120px, 22vw, 160px);
    height: auto;
    margin-bottom: clamp(0.8rem, 2vh, 1.2rem);
    transition: all 0.4s ease;
    filter: drop-shadow(0 8px 24px rgba(244, 180, 0, 0.3));
    animation: logo-resplendor 3s ease-in-out infinite;
    display: block;
    margin-left: auto;
    margin-right: auto;
}

@keyframes logo-resplendor {
    0%, 100% {
        filter: drop-shadow(0 8px 24px rgba(244, 180, 0, 0.3));
        transform: scale(1);
    }
}

```

```

    }
    50% {
      filter: drop-shadow(0 12px 32px rgba(244, 180, 0, 0.5));
      transform: scale(1.02);
    }
  }

.logo:hover {
  transform: scale(1.05) rotate(2deg);
  filter: drop-shadow(0 16px 40px rgba(244, 180, 0, 0.6));
}

/* -----
   Header de la tarjeta
   Título principal y línea decorativa con animación. Usar h2 para SEO
   y accesibilidad, y mantener contraste suficiente con el fondo.
*/
/* ⚡ HEADER ELEGANTE */
.header {
  text-align: center;
  margin-bottom: clamp(1.6rem, 3.5vh, 2.2rem);
  padding-bottom: clamp(1rem, 2.5vh, 1.5rem);
  position: relative;
}

.header::after {
  content: "";
  position: absolute;
  bottom: 0;
  left: 50%;
  transform: translateX(-50%);
  width: clamp(80px, 18vw, 120px);
  height: clamp(3px, 0.6vh, 4px);
  background: var(--gradiente-dorado);
  border-radius: 2px;
  box-shadow: var(--resplandor-dorado);
  animation: linea-pulso 2s ease-in-out infinite;
}

@keyframes linea-pulso {
  0%, 100% { width: clamp(80px, 18vw, 120px); opacity: 0.8; }
  50% { width: clamp(100px, 22vw, 140px); opacity: 1; }
}

.header h2 {
  color: var(--negro);
  font-size: clamp(1.6rem, 4vw, 2rem);
  margin-bottom: clamp(0.5rem, 1.5vh, 0.7rem);
}

```

```

font-weight: 700;
letter-spacing: -0.4px;
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
text-shadow: 0 2px 4px rgba(0, 0, 0, 0.05);
background: var(--gradiente-dorado);
-webkit-background-clip: text;
background-clip: text;
-webkit-text-fill-color: transparent;
animation: titulo-brillo 2s ease-in-out infinite;
line-height: 1.15;
}

@keyframes titulo-brillo {
  0%, 100% { opacity: 0.95; transform: scale(1); }
  50% { opacity: 1; transform: scale(1.005); }
}

.subtitle {
  color: var(--gris-oscuro);
  font-size: clamp(0.85rem, 2.5vw, 1rem);
  font-weight: 400;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  text-transform: uppercase;
  letter-spacing: clamp(0.5px, 0.1vw, 0.8px);
  opacity: 0.8;
  margin-bottom: clamp(0.4rem, 1vh, 0.6rem);
  line-height: 1.3;
}

/*
-----  

  Formulario ` .login-form` y grupos de control  

` .form-group` y ` .input-group` organizan inputs/labels. Validaciones  

y estados visuales se controlan con clases (.success, .error).
*/
/* FORMULARIO */
.login-form {
  width: 100%;
  position: relative;
}

/* GRUPOS DE FORMULARIO ELEGANTES */
.form-group {
  margin-bottom: clamp(1.2rem, 2.6vh, 1.5rem);
  position: relative;
}

.form-group label {
  display: block;
}

```

```

margin-bottom: clamp(0.5rem, 1.2vh, 0.7rem);
color: var(--amarillo);
font-weight: 600;
font-size: clamp(0.85rem, 2.2vw, 0.95rem);
text-transform: uppercase;
letter-spacing: clamp(0.3px, 0.08vw, 0.5px);
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
text-shadow: 0 0 10px rgba(244, 180, 0, 0.3);
}

/*
----- Inputs y placeholders
` .input-field` y reglas de focus proporcionan feedback visual. Si
añades
    nuevos inputs, reutiliza estas clases para mantener consistencia.
*/
/* 💡 INPUT FIELDS ELEGANTES */
.input-group {
  position: relative;
  margin-bottom: clamp(1.2rem, 2.6vh, 1.5rem);
}

.input-label {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  font-weight: 600;
  color: var(--amarillo);
  font-size: clamp(0.85rem, 2.2vw, 0.95rem);
  display: block;
  margin-bottom: clamp(0.5rem, 1.2vh, 0.7rem);
  text-transform: uppercase;
  letter-spacing: clamp(0.3px, 0.08vw, 0.5px);
  text-shadow: 0 0 10px rgba(244, 180, 0, 0.3);
}

.input-field {
  width: 100%;
  padding: clamp(0.9rem, 2.2vh, 1.1rem) clamp(1rem, 2.5vw, 1.2rem);
  border: 2px solid var(--gris-claro);
  border-radius: clamp(1rem, 2vh, 1.4rem);
  font-size: clamp(0.95rem, 2.2vw, 1.05rem);
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  color: var(--negro);
  background: rgba(255, 255, 255, 0.98);
  transition: all 0.4s cubic-bezier(0.25, 0.46, 0.45, 0.94);
  outline: none;
  backdrop-filter: blur(5px);
  position: relative;
  box-sizing: border-box;
}

```

```

}

/* Estilos para form-group input también - MÁS ESTRECHOS */
.form-group input {
  width: 100%;
  padding: clamp(0.9rem, 2.2vh, 1.1rem) clamp(1rem, 2.5vw, 1.2rem);
  border: 2px solid var(--gris-claro);
  border-radius: clamp(1rem, 2vh, 1.4rem);
  font-size: clamp(0.95rem, 2.2vw, 1.05rem);
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  color: var(--negro);
  background: rgba(255, 255, 255, 0.98);
  transition: all 0.4s cubic-bezier(0.25, 0.46, 0.45, 0.94);
  outline: none;
  backdrop-filter: blur(5px);
  box-sizing: border-box;
}

.input-field:focus,
.form-group input:focus {
  border-color: var(--amarillo);
  box-shadow:
    0 0 0 clamp(3px, 0.8vh, 5px) rgba(244, 180, 0, 0.15),
    var(--resplandor-dorado);
  transform: translateY(clamp(-1px, -0.3vh, -2px)) scale(1.005);
  background: var(--blanco);
}

.input-field::placeholder,
.form-group input::placeholder {
  color: var(--gris-oscuro);
  opacity: 0.7;
  font-weight: 400;
  text-transform: uppercase;
  letter-spacing: clamp(0.3px, 0.08vw, 0.5px);
}

/*
-----  

  Botón principal ` .btn-primary`  

  Estilos del CTA. Mantener interacción hover/active y evitar cambios de  

  anchura dinámicos que rompan el layout.
*/
/* 🔍 BOTÓN MÁS ESTRECHO - PROPORCIÓN PERFECTA */
.btn-primary {
  width: 100%;
  padding: clamp(1rem, 2.5vh, 1.25rem) clamp(1.2rem, 2.8vw, 1.5rem);
  background: var(--gradiente-dorado);
  color: var(--negro);
}

```

```

border: none;
border-radius: clamp(1rem, 2vh, 1.4rem);
font-size: clamp(0.95rem, 2.2vw, 1.05rem);
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
font-weight: 700;
text-transform: uppercase;
letter-spacing: clamp(0.5px, 0.1vw, 0.8px);
cursor: pointer;
transition: all 0.4s cubic-bezier(0.25, 0.46, 0.45, 0.94);
box-shadow: var(--sombra-magica);
position: relative;
overflow: hidden;
margin-top: clamp(0.8rem, 2vh, 1.2rem);
box-sizing: border-box;
}

.btn-primary::before {
  content: "";
  position: absolute;
  top: 0;
  left: -100%;
  width: 100%;
  height: 100%;
  background: linear-gradient(90deg, transparent, rgba(249, 196, 31, 0.2), transparent);
  transition: left 0.5s ease;
}

.btn-primary:hover::before {
  left: 100%;
}

.btn-primary:hover {
  background: linear-gradient(135deg, var(--amarillo), var(--dorado));
  color: var(--negro);
  transform: translateY(clamp(-0.5px, -0.2vh, -1px)) scale(1.005);
  box-shadow: 0 8px 24px rgba(249, 196, 31, 0.3);
}

.btn-primary:active {
  transform: translateY(0);
  box-shadow: 0 4px 16px rgba(0, 0, 0, 0.2);
}

/* -----
   Responsive y media queries
   Reglas por breakpoints para móviles y tablets. Se desactivan
   animaciones

```

```

    y partículas en pantallas pequeñas para mejorar rendimiento.
*/
/* ⚡ DISEÑO UNIVERSAL - FUNCIONA EN CUALQUIER ZOOM Y PANTALLA */
/* Sin media queries complejas - Solo solución universal */

/* Solo para pantallas extremadamente pequeñas (móviles muy antiguos) */
@media (max-width: 280px) {
    body {
        padding: 4px;
    }

    .login-card {
        padding: min(1.2rem, 3vh) min(1rem, 2vw);
    }

    .header h2 {
        font-size: min(1.4rem, 4vw, 3vh);
    }

    .logo {
        width: min(100px, 18vw, 12vh);
    }

    .logo-bastos {
        width: clamp(25px, 6vw, 35px) !important;
        height: clamp(25px, 6vw, 35px) !important;
    }
}

/* 📱 OPTIMIZACIÓN MÍNIMA PARA TODOS LOS MÓVILES Y TABLETS */
@media (max-width: 1024px) {
    body {
        align-items: center !important;
        justify-content: center !important;
        padding: clamp(4px, 1.2vh, 10px) !important;
        min-height: 100vh !important;
        height: 100vh !important;
    }

    .login-container {
        max-height: none !important;
        margin: 0 auto !important;
        position: relative !important;
        top: 0 !important;
        transform: none !important;
        display: flex !important;
        flex-direction: column !important;
        justify-content: center !important;
    }
}

```

```

    align-items: center !important;
    /* Optimización para tablets grandes */
    max-width: clamp(320px, 75vw, 450px) !important;
}

.login-card {
    margin: 0 !important;
    padding: clamp(1.8rem, 4vh, 2.6rem) clamp(1.4rem, 3.2vw, 1.9rem)
!important;
    max-height: none !important;
}
}

/* 📱 MÓVILES MEDIANOS Y PEQUEÑOS (hasta 768px) */
@media (max-width: 768px) {
    body {
        padding: clamp(4px, 0.8vh, 8px) !important;
        /* 🕒 FONDO BLANCO PURO PARA MÓVILES - SIN TEXTURAS */
        background: #ffffff !important;
        background-color: #ffffff !important;
        background-image: none !important;
        /* Desactivar partículas en móviles */
        /* ⚡ COMPENSAR BARRA DEL NAVEGADOR - SUBIR LOGIN */
        align-items: center !important;
        justify-content: center !important;
        /* 💸 Compensación moderada para mantener diseño */
        padding-bottom: clamp(30px, 4vh, 50px) !important;
    }
}

/* 🔍 OCULTAR PARTÍCULAS EN MÓVILES */
body::before,
body::after {
    display: none !important;
}

.login-container {
    max-width: clamp(300px, 88vw, 400px) !important;
    max-height: none !important;
    margin: 0 auto !important;
    /* ↑ Transformación ligera para mantener elegancia */
    transform: translateY(-8px) !important;
    position: relative !important;
    top: 0 !important;
}

.login-card {
    padding: clamp(1.4rem, 3.2vh, 2.4rem) clamp(1.2rem, 2.8vw, 1.7rem)
!important;
}

```

```

/* Fondo más sólido en móviles */
background: rgba(255, 255, 255, 0.98) !important;
backdrop-filter: none !important;
-webkit-backdrop-filter: none !important;
max-height: none !important;
}

.header {
  margin-bottom: clamp(1.2rem, 2.8vh, 1.8rem) !important;
  padding-bottom: clamp(0.8rem, 1.8vh, 1.2rem) !important;
}

.header h2 {
  font-size: clamp(1.6rem, 5vw, 2rem) !important;
  margin-bottom: clamp(0.4rem, 1.2vh, 0.6rem) !important;
}

.form-group {
  margin-bottom: clamp(1rem, 2.2vh, 1.3rem) !important;
}

.input-group {
  margin-bottom: clamp(1rem, 2.2vh, 1.3rem) !important;
}

.logo {
  width: clamp(100px, 20vw, 140px) !important;
  margin-bottom: clamp(0.8rem, 1.8vh, 1.2rem) !important;
}

.logo-bastos {
  width: clamp(30px, 7vw, 40px) !important;
  height: clamp(30px, 7vw, 40px) !important;
  margin-bottom: clamp(0.3rem, 0.8vh, 0.5rem) !important;
}

/*
  ☺ MÓVILES PEQUEÑOS (hasta 480px)
*/
@media (max-width: 480px) {
  body {
    padding: clamp(3px, 0.6vh, 6px) !important;
    /* ☺ FONDO BLANCO PURO Y LIMPIO PARA MÓVILES PEQUEÑOS */
    background: #ffffff !important;
    background-color: #ffffff !important;
    background-image: none !important;
    /* ☺ COMPENSAR BARRA DEL NAVEGADOR EN MÓVILES PEQUEÑOS */
    align-items: center !important;
    justify-content: center !important;
  }
}

```

```

/* ⚡ Compensación moderada para mantener diseño */
padding-bottom: clamp(25px, 3vh, 45px) !important;
}

/* ✎ ASEGURAR QUE NO HAY PARTÍCULAS */
body::before,
body::after {
    display: none !important;
    opacity: 0 !important;
}

.login-container {
    max-width: clamp(280px, 94vw, 360px) !important;
    max-height: none !important;
    margin: 0 auto !important;
    /* ↑ Transformación suave para mantener elegancia */
    transform: translateY(-5px) !important;
    position: relative !important;
    top: 0 !important;
}

.login-card {
    padding: clamp(1.2rem, 2.8vh, 2rem) clamp(1rem, 2.2vw, 1.5rem)
!important;
    background: #ffffff !important;
    backdrop-filter: none !important;
    -webkit-backdrop-filter: none !important;
    box-shadow:
        0 4px 20px rgba(0, 0, 0, 0.08),
        0 2px 8px rgba(244, 180, 0, 0.05) !important;
    max-height: none !important;
}

.header {
    margin-bottom: clamp(1rem, 2.4vh, 1.6rem) !important;
    padding-bottom: clamp(0.6rem, 1.4vh, 1rem) !important;
}

.header h2 {
    font-size: clamp(1.5rem, 5.2vw, 1.9rem) !important;
    margin-bottom: clamp(0.3rem, 1vh, 0.5rem) !important;
}

.form-group {
    margin-bottom: clamp(0.9rem, 2vh, 1.2rem) !important;
}

.input-group {

```

```

        margin-bottom: clamp(0.9rem, 2vh, 1.2rem) !important;
    }

.form-group label,
.input-label {
    margin-bottom: clamp(0.4rem, 1vh, 0.6rem) !important;
    font-size: clamp(0.82rem, 2.1vw, 0.92rem) !important;
}

.input-field,
.form-group input {
    padding: clamp(0.85rem, 2vh, 1.05rem) clamp(0.9rem, 2.2vw, 1.1rem)
!important;
    font-size: max(16px, clamp(0.92rem, 2.2vw, 1.02rem)) !important;
}

.btn-primary {
    padding: clamp(0.9rem, 2.2vh, 1.15rem) clamp(1.1rem, 2.4vw, 1.3rem)
!important;
    font-size: max(16px, clamp(0.92rem, 2.2vw, 1.02rem)) !important;
    margin-top: clamp(0.6rem, 1.4vh, 1rem) !important;
}

.logo {
    width: clamp(95px, 19vw, 125px) !important;
    margin-bottom: clamp(0.6rem, 1.4vh, 1rem) !important;
}

.logo-bastos {
    width: clamp(28px, 6.5vw, 38px) !important;
    height: clamp(28px, 6.5vw, 38px) !important;
    margin-bottom: clamp(0.2rem, 0.6vh, 0.4rem) !important;
}
}

/* ☺ MÓVILES MUY PEQUEÑOS (hasta 320px) */
@media (max-width: 320px) {
    .login-container {
        max-width: 96vw !important;
    }

    .login-card {
        padding: clamp(1rem, 2.4vh, 1.8rem) clamp(0.8rem, 1.8vw, 1.3rem)
!important;
    }

    .header h2 {
        font-size: clamp(1.3rem, 5.8vw, 1.7rem) !important;
    }
}

```

```

        }

/* ☺ SOLO PANTALLAS MUY PEQUEÑAS EN ALTURA - OPTIMIZACIÓN MÍNIMA */
@media (max-height: 640px) {
    .login-card {
        padding: clamp(1.2rem, 2.5vh, 2rem) clamp(1.2rem, 2.8vw, 1.8rem)
!important;
        max-height: none !important;
    }

.header {
    margin-bottom: clamp(1rem, 2.2vh, 1.6rem) !important;
    padding-bottom: clamp(0.6rem, 1.4vh, 1rem) !important;
}

.form-group {
    margin-bottom: clamp(0.9rem, 1.8vh, 1.2rem) !important;
}

.input-group {
    margin-bottom: clamp(0.9rem, 1.8vh, 1.2rem) !important;
}
}

/* Solo para pantallas muy altas (monitores verticales) */
@media (min-aspect-ratio: 1/2) {
    .login-container {
        max-height: 85vh;
    }
}

/* Solo para pantallas muy anchas (monitores ultrawide) */
@media (min-aspect-ratio: 3/1) {
    .login-container {
        max-width: min(350px, 25vw);
    }
}

/* 🔎 VALIDACIONES Y ESTADOS */
/* -----
   Validaciones visuales
   Estados CSS para inputs válidos/invalidos y mensajes de error. JS solo
   alterna clases o atributos (no manipular estilos directamente si se
   puede).
*/
/* 🔎 VALIDACIONES Y ESTADOS */
.input-field:valid:not(:placeholder-shown),

```

```
.form-group input:valid:not(:placeholder-shown) {  
    border-color: #28a745;  
    box-shadow: 0 0 0 4px rgba(40, 167, 69, 0.1);  
}  
  
.input-field:invalid:not(:placeholder-shown),  
.form-group input:invalid:not(:placeholder-shown) {  
    border-color: #dc3545;  
    box-shadow: 0 0 0 4px rgba(220, 53, 69, 0.1);  
}  
  
/* 🚨 MENSAJES DE ERROR */  
.error-message {  
    color: #dc3545;  
    font-size: 0.8rem;  
    margin-top: 12px;  
    display: block;  
    opacity: 0;  
    transition: all 0.4s ease;  
    font-weight: 600;  
    padding: 8px 12px;  
    background: rgba(220, 53, 69, 0.1);  
    border-radius: 8px;  
    border-left: 4px solid #dc3545;  
}  
  
.error-message.show {  
    opacity: 1;  
    transform: translateY(0px);  
}  
  
/* 🔍 ESTADOS DE CARGA */  
.btn-primary:disabled {  
    background: linear-gradient(135deg, var(--gris-claro), var(--gris));  
    color: rgba(255, 255, 255, 0.6);  
    cursor: not-allowed;  
    transform: none;  
    box-shadow: none;  
}  
  
.spinner {  
    width: 20px;  
    height: 20px;  
    border: 2px solid transparent;  
    border-top: 2px solid var(--amarillo);  
    border-radius: 50%;  
    animation: spin 1s linear infinite;  
    display: none;  
}
```

```

    margin-left: 10px;
}

@keyframes spin {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}

/* Animaciones de entrada - Modernizadas */
@keyframes fadeInUp {
  from {
    opacity: 0;
    transform: translateY(40px) scale(0.95);
  }
  to {
    opacity: 1;
    transform: translateY(0) scale(1);
  }
}

.login-card {
  animation: fadeInUp 0.8s cubic-bezier(0.25, 0.46, 0.45, 0.94);
}

/* Estados de validación visual - Modernizados */
.form-group.success input {
  border-color: #28a745;
  background: linear-gradient(135deg, #f8ffff, #ffffff);
}

.form-group.error input {
  border-color: #dc3545;
  background: linear-gradient(135deg, #fff8f8, #ffffff);
}

.form-group.success input:focus {
  box-shadow: 0 0 0 4px rgba(40, 167, 69, 0.15);
}

.form-group.error input:focus {
  box-shadow: 0 0 0 4px rgba(220, 53, 69, 0.15);
}

/* ===== AJUSTES FINALES MODERNOS PARA UNIFICAR ESTILO ===== */

/* Header de tarjeta - Modernizado */
.card-header {
  background: linear-gradient(135deg, var(--negro), #2a2a2a) !important;
}

```

```

color: var(--amarillo) !important;
font-weight: 700 !important;
font-size: 1.2rem !important;
border-radius: 16px 16px 0 0 !important;
padding: 1.5rem !important;
position: relative;
overflow: hidden;
}

.card-header::before {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  height: 3px;
  background: linear-gradient(90deg, var(--amarillo), var(--naranja));
}

/* Botón actualizar pequeño - Modernizado */
.btn-sm.btn-outline-light {
  padding: 8px 16px !important;
  font-size: 0.85rem !important;
  font-weight: 600 !important;
  border-radius: 8px !important;
  border: 2px solid var(--amarillo) !important;
  color: var(--amarillo) !important;
  background: transparent !important;
  transition: all 0.3s ease !important;
}

.btn-sm.btn-outline-light:hover {
  background: var(--amarillo) !important;
  color: var(--negro) !important;
  transform: translateY(-2px) !important;
  box-shadow: 0 4px 12px rgba(249, 196, 31, 0.3) !important;
}

/* Card de tabla sin padding extra */
.card-body.p-0 {
  padding: 0 !important;
  border-radius: 0 0 16px 16px !important;
}

/* Ajuste visual para tabla responsive */
.table-responsive {
  width: 100% !important;
  overflow-x: auto !important;
}

```

```

padding: 1.5rem !important;
border-radius: 0 0 16px 16px !important;
}

/* Tabla injectada dinámicamente */
#usuarioLista table {
    min-width: 700px !important;
    white-space: nowrap !important;
    border-radius: 12px !important;
    overflow: hidden !important;
    box-shadow: var(--sombra-moderna) !important;
}

/* Líneas decorativas responsivas y CENTRADO UNIVERSAL MÓVIL */
@media (max-width: 1024px) {
    .decor-line {
        display: none !important;
    }

    .card-header {
        padding: 1rem !important;
        font-size: 1.1rem !important;
    }

    .table-responsive {
        padding: 1rem !important;
    }
}

/* 📱 CENTRADO ELEGANTE PARA TODOS LOS DISPOSITIVOS MÓVILES */
body {
    padding: clamp(3px, 1.2vh, 10px) !important;
    display: flex !important;
    align-items: center !important;
    justify-content: center !important;
    min-height: 100vh !important;
    overflow-x: hidden !important;
    /* 💡 COMPENSACIÓN MODERADA PARA MANTENER ELEGANCIA */
    padding-bottom: clamp(25px, 3vh, 45px) !important;

    /* 💡 MÓVILES: FONDO BLANCO PURO SIN TEXTURAS */
    background: #ffffff !important;
    background-color: #ffffff !important;
    background-image: none !important;
    -webkit-background-size: auto !important;
    background-size: auto !important;
    background-attachment: scroll !important;
    background-repeat: no-repeat !important;
}

```

```

/* ⚡ DESACTIVAR PARTÍCULAS EN TODOS LOS MÓVILES */
body::before,
body::after {
    display: none !important;
}

.login-container {
    position: relative !important;
    top: 0 !important;
    transform: translateY(-5px) !important;
    margin: 0 auto !important;
    display: flex !important;
    flex-direction: column !important;
    justify-content: center !important;
    align-items: center !important;
    height: auto !important;
    max-height: 96vh !important;
    width: 100% !important;
}

.login-card {
    margin: 0 !important;
    width: 100% !important;
    box-sizing: border-box !important;
    max-height: 94vh !important;
}

/* Optimización específica para inputs en móviles */
.input-field,
.form-group input {
    font-size: max(16px, clamp(0.95rem, 2.2vw, 1.05rem)) !important;
    /* Evita zoom en iOS */
}

.btn-primary {
    font-size: max(16px, clamp(0.95rem, 2.2vw, 1.05rem)) !important;
    padding: clamp(1rem, 2.8vh, 1.4rem) clamp(1.2rem, 2.5vw, 1.5rem)
!important;
}

/* 📱 ORIENTACIÓN LANDSCAPE EN MÓVILES - MANTENER ELEGANCIA */
@media (max-width: 768px) and (orientation: landscape) {
    body {
        padding: 2px !important;
        padding-bottom: 8px !important;
    }
}

```

```

.login-container {
  max-height: 96vh !important;
  transform: translateY(-3px) !important;
}

.login-card {
  padding: clamp(1rem, 2.2vh, 1.6rem) clamp(1rem, 2.4vw, 1.5rem)
!important;
  max-height: 93vh !important;
  overflow-y: auto !important;
}

.header {
  margin-bottom: clamp(0.8rem, 1.8vh, 1.2rem) !important;
  padding-bottom: clamp(0.5rem, 1.2vh, 0.8rem) !important;
}

.header h2 {
  font-size: clamp(1.3rem, 4.2vw, 1.7rem) !important;
  margin-bottom: clamp(0.2rem, 0.6vh, 0.4rem) !important;
}

.form-group {
  margin-bottom: clamp(0.7rem, 1.6vh, 1.1rem) !important;
}

.input-group {
  margin-bottom: clamp(0.7rem, 1.6vh, 1.1rem) !important;
}

.logo {
  width: clamp(75px, 12vw, 105px) !important;
  margin-bottom: clamp(0.4rem, 1vh, 0.8rem) !important;
}

.logo-bastos {
  width: clamp(22px, 5vw, 32px) !important;
  height: clamp(22px, 5vw, 32px) !important;
  margin-bottom: clamp(0.15rem, 0.4vh, 0.3rem) !important;
}
}

```

| | └— Manual.css

```

/* Manual.css – documentación in-situ
Propósito: estilos para las páginas de manual/guias internas.
Resumen:
```

```

- Variables en :root (colores y sombras) para tema consistente.
- .guia-container: contenedor principal de las guías.
- .guia-card / .guia-header / .guia-body: componentes reutilizables.
- Se incluye responsive para tablet/móvil y pequeñas optimizaciones de sidebar.
*/
:root {
    --amarillo: #f4b400;           /* Dorado intenso premium */
    --amarillo-claro: #fff8e1;
    --amarillo-texto: #f4d03f;
    --negro: #000000;
    --blanco: #ffffff;
    --gris: #2a2a2a;
    --gris-fondo: #f8f9fa;
    --sombra-elevada: 0 16px 48px rgba(0, 0, 0, 0.15);
    --sombra: rgba(0, 0, 0, 0.4);
}

/* Contenedor principal - Ultra Premium */
/* ===== */
    Contenedor principal (.guia-container)
    Estilo del contenedor que envuelve las guías. Ajusta padding y border-radius
        aquí para cambiar la sensación de espacio. Mantener backdrop-filter si
        se
            desea efecto translúcido (podría impactar rendimiento en móviles).
/*
.guia-container {
    background: linear-gradient(135deg, rgba(255, 255, 255, 0.98) 0%,
    rgba(248, 249, 250, 0.95) 100%);
    padding: 48px;
    border-radius: 24px;
    box-shadow: var(--sombra-elevada);
    border: 1px solid rgba(244, 180, 0, 0.1);
    backdrop-filter: blur(20px);
    position: relative;
    overflow: hidden;
}

.guia-container::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 6px;
    background: linear-gradient(90deg, var(--amarillo) 0%, #ffd700 50%,
    #ffed4e 100%);
}

```

```

    z-index: 1;
}

/* Título principal - Premium */
.guia-titulo {
  color: var(--amarillo);
  font-weight: 800;
  font-size: 2.5rem;
  text-align: center;
  text-transform: uppercase;
  letter-spacing: 2px;
  margin-bottom: 2rem;
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.1);
  position: relative;
  padding-bottom: 20px;
}

.guia-titulo::after {
  content: "";
  position: absolute;
  bottom: 0;
  left: 50%;
  transform: translateX(-50%);
  width: 100px;
  height: 4px;
  background: linear-gradient(90deg, var(--amarillo), #ffd700);
  border-radius: 2px;
}

/* Tarjetas de guía - Ultra Premium */
/* ===== */
Tarjetas de guía (.guia-card)
Elementos individuales con borde dorado y pequeño efecto decorativo; se animan al hover. Reutilízalas para secciones independientes del manual.
*/
.guia-card {
  background: rgba(255, 255, 255, 0.98);
  border: 2px solid var(--amarillo);
  border-left: 8px solid var(--amarillo);
  border-radius: 20px;
  padding: 32px;
  transition: all 0.4s ease;
  box-shadow: 0 8px 32px rgba(0, 0, 0, 0.08);
  backdrop-filter: blur(10px);
  position: relative;
  overflow: hidden;
  margin-bottom: 24px;
}

```

```
.guia-card::before {
  content: "";
  position: absolute;
  top: 0;
  right: 0;
  width: 60px;
  height: 60px;
  background: linear-gradient(135deg, var(--amarillo) 0%, #ffd700 100%);
  border-radius: 0 0 60px;
  opacity: 0.1;
  transition: all 0.3s ease;
}

.guia-card:hover {
  transform: translateY(-8px) scale(1.02);
  box-shadow: var(--sombra-elevada);
  border-color: #ffd700;
}

.guia-card:hover::before {
  opacity: 0.2;
  width: 80px;
  height: 80px;
}

/* Headers de sección - Premium */
/* -----
   Cabecera de sección (.guia-header)
   Títulos de pequeñas secciones dentro de cada tarjeta. Incluye icono
   decorativo acompañado de texto en mayúsculas.
*/
.guia-header {
  font-size: 1.4rem;
  font-weight: 700;
  color: var(--amarillo);
  margin-bottom: 16px;
  display: flex;
  align-items: center;
  white-space: nowrap;
  text-transform: uppercase;
  letter-spacing: 1px;
}

.guia-header i {
  margin-right: 16px;
  font-size: 1.6rem;
  padding: 12px;
```

```

background: linear-gradient(135deg, var(--amarillo) 0%, #ffd700 100%);
color: var(--negro);
border-radius: 12px;
box-shadow: 0 4px 16px rgba(244, 180, 0, 0.3);
}

/* Contenido del cuerpo - Premium */
/* -----
   Cuerpo del contenido (.guia-body)
   Texto principal de las tarjetas. Ajustar line-height y font-size aquí
   cuando quieras mejorar legibilidad en móviles o impresiones.
*/
.guia-body {
  color: var(--gris);
  font-size: 1.1rem;
  line-height: 1.8;
  font-weight: 400;
  padding-left: 8px;
}

/* Imagen de header - Premium */
.img-header {
  width: 120px;
  height: auto;
  max-width: 100%;
  object-fit: contain;
  display: block;
  margin: 0 auto 24px;
  border-radius: 16px;
  box-shadow: 0 8px 32px rgba(0, 0, 0, 0.1);
  transition: all 0.3s ease;
}

.img-header:hover {
  transform: scale(1.05);
  box-shadow: 0 12px 48px rgba(244, 180, 0, 0.2);
}

/* ====== RESPONSIVE ===== */
/* Tablets ≥576 px - ≤991 px (iPad vertical, tablets medianas) */
/* ======
   Responsive: tablets y móviles
   Ajustes por breakpoint para mantener la legibilidad y reducir
   decoraciones
   en pantallas pequeñas.
*/

```

```

@media (max-width: 991.98px) {
  .guia-container {
    padding: 32px; /* Menos relleno */
    border-radius: 18px;
  }

  .guia-titulo {
    font-size: 2rem; /* Reducir título */
    margin-bottom: 1.5rem;
  }

  .guia-card {
    padding: 24px; /* Cartas más compactas */
    border-left-width: 6px;
    border-radius: 16px;
  }

  .guia-header {
    font-size: 1.25rem;
    margin-bottom: 12px;
  }

  .guia-header i {
    font-size: 1.4rem;
    padding: 10px;
  }

  .guia-body {
    font-size: 1rem;
    line-height: 1.6;
  }

  .img-header {
    width: 90px;
    margin-bottom: 20px;
  }
}

/* Teléfonos ≤575 px (modo móvil) */
/* -----
   Responsive: móviles pequeños
   Rellenos reducidos y animaciones más sutiles para mejorar UX en
móviles.
*/
@media (max-width: 575.98px) {
  .guia-container {
    padding: 20px; /* Relleno mínimo */
    border-radius: 14px;
}

```

```

}

.guias-titulo {
  font-size: 1.6rem;
  letter-spacing: 1px;
  padding-bottom: 14px;
}

.guias-titulo::after {
  width: 70px; /* Línea decorativa más corta */
}

.guias-card {
  padding: 20px;
  margin-bottom: 20px;
  border-left-width: 5px;
  border-radius: 14px;
}

.guias-card:hover {
  transform: translateY(-4px) scale(1.01); /* Animación más sutil */
}

.guias-header {
  font-size: 1.1rem;
  letter-spacing: 0.5px;
}

.guias-header i {
  font-size: 1.2rem;
  padding: 8px;
}

.guias-body {
  font-size: 0.95rem;
  line-height: 1.5;
  padding-left: 0; /* Ajustar sangría */
}

}

/* Sidebar: altura estable y logo como Bitácora */
/* -----
Ajustes de sidebar usados en la vista del manual
(altura estable, logo compacto). Estos estilos son pequeños tweaks para
integrar la guía en la interfaz principal.
*/
.sidebar-header{

```

```

box-sizing: border-box;
height: 80px; /* o 79.52px si quieres clavarlo */
padding: 12px 12px;
display: flex;
align-items: center;
gap: 12px;
position: sticky;
top: 0;
}

.sidebar-header .img-header{
width: 43.4px;
height: 44.3px;
object-fit: contain;
margin: 0;
max-width: none;
transform: none;
}
.sidebar-header .img-header:hover{ transform:none; box-shadow:none; }

.sidebar-header h3 {
font-size: 1.25rem;
margin: 0;
font-weight: 700;
color: var(--amarillo);
text-shadow: 0 1px 2px rgba(0, 0, 0, 0.3);
}
.img-header:hover {
transform: scale(1.1) rotate(5deg);
box-shadow: 0 4px 16px rgba(249, 196, 31, 0.3);
}
.img-header {
width: 44px;
height: 44px;
object-fit: contain;
border-radius: 8px;
background: rgba(249, 196, 31, 0.2);
padding: 6px;
transition: all 0.3s ease;
box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
}
img, svg {
vertical-align: middle;
}
*, ::after, ::before {
box-sizing: border-box;
}

```

```
.sidebar-header::before {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 2px;
  background: linear-gradient(90deg, var(--amarillo) 0%, #ffd700 100%);
}
```

## | | └ Manual2.css

```
/* Manual2.css – documentación in-situ
Propósito: estilos para la segunda versión de las páginas de
manual/guias.

/* === Variables globales (:root) ===
Define colores y sombras reutilizables. Cambia valores aquí para
aplicar
tema consistente en todas las guías.
*/
:root {
  --amarillo: #f4b400;
  --amarillo-claro: #fff8e1;
  --amarillo-texto: #f4d03f;
  --negro: #000000;
  --blanco: #ffffff;
  --gris: #2a2a2a;
  --gris-fondo: #f8f9fa;
  --sombra-elevada: 0 16px 48px rgba(0, 0, 0, 0.15);
}

/* === Contenedor principal (.guia-container) ===
Contenedor centrado que envuelve el contenido de la guía. Ajusta
`padding`, `max-width` y `min-height` aquí para controlar la proporción
visual en pantallas grandes y pequeñas.
*/
.guia-container {
  background: linear-gradient(135deg, rgba(255, 255, 255, 0.98) 0%,
  rgba(248, 249, 250, 0.95) 100%);
  padding: 48px;
  border-radius: 24px;
  box-shadow: var(--sombra-elevada);
  border: 1px solid rgba(244, 180, 0, 0.1);
  position: relative;
  overflow: hidden;
  overflow-x: hidden;

  /* ↩ Más angosto + más alto */
```

```

max-width: 900px;      /* ancho como lo tenías */
min-height: 85vh;       /* NUEVO: más alto para mejor proporción */
margin-left: auto;
margin-right: auto;
}

.guia-container::before {
  content: "";
  position: absolute;
  top: 0; left: 0;
  width: 100%; height: 6px;
  background: linear-gradient(90deg, var(--amarillo) 0%, #ffd700 50%, #ffed4e 100%);
  z-index: 1;
}

/* === Título de la guía (.guia-titulo) ===
   Estilo del encabezado principal: usar mayúsculas y una línea decorativa
   debajo para consistencia visual con el tema dorado.
*/
.guia-titulo {
  color: var(--amarillo);
  font-weight: 800;
  font-size: 2.2rem;
  text-transform: uppercase;
  letter-spacing: 2px;
  margin-bottom: 2rem;
  position: relative;
  padding-bottom: 16px;
  text-align: center;
}

.guia-titulo::after {
  content: "";
  position: absolute;
  bottom: 0; left: 50%;
  transform: translateX(-50%);
  width: 100px; height: 4px;
  background: linear-gradient(90deg, var(--amarillo), #ffd700);
  border-radius: 2px;
}

/* === Acordeón (.guia-accordion) ===
   Implementa una lista de `details/summary` estilizada. Las reglas
   controlan
   el resumen (header) y el contenido expandible con transiciones suaves.
*/
.guia-accordion {

```

```
display: grid;
gap: 14px;
}

.acc-item {
background: #fff;
border: 1px solid rgba(244, 180, 0, 0.25);
border-radius: 12px;
box-shadow: 0 6px 18px rgba(0, 0, 0, 0.06);
overflow: hidden;
}

.acc-item[open] {
box-shadow: 0 10px 28px rgba(0, 0, 0, 0.08);
}

.acc-summary {
list-style: none;
cursor: pointer;
display: flex;
align-items: center;
justify-content: space-between;
gap: 12px;
padding: 14px 16px;
font-weight: 700;
color: var(--negro);
background: rgba(249, 196, 31, 0.08);
}

.acc-summary::-webkit-details-marker { display: none; }

.acc-summary > span {
display: inline-flex;
align-items: center;
gap: 10px;
line-height: 1.2;
}

.acc-summary .bi { font-size: 1.1rem; }

.acc-arrow {
transition: transform 0.2s ease;
font-size: 1rem;
}

.acc-item[open] > .acc-summary .acc-arrow { transform: rotate(180deg); }

/* Transición suave al abrir/cerrar */
```

```

.acc-content {
  max-height: 0;
  overflow: hidden;
  padding: 0 16px;
  border-top: 0;
  transition: max-height .35s ease, padding .2s ease, border-top-color
  .2s ease;
  will-change: max-height;
}

.acc-item[open] > .acc-content {
  max-height: 1200px;
  padding: 12px 16px 16px;
  border-top: 1px dashed rgba(249, 196, 31, 0.45);
}

/* === Acordeón anidado (videos) ===
   Variante para contenidos anidados (por ejemplo listas de videos). Tiene
   fondo/colores diferenciados y límites de altura cuando está abierto.
*/
.acc-item.nested {
  margin: 10px 10px 0 10px;
  background: #ffffdf5;
  border-color: rgba(249, 196, 31, 0.35);
}
.acc-item.nested .acc-summary {
  font-weight: 600;
  font-size: 1rem;
  background: rgba(249, 196, 31, 0.07);
}
.acc-item.nested .acc-content {
  max-height: 0;
  overflow: hidden;
  padding: 0 14px;
  border-top: 0;
  transition: max-height .3s ease, padding .2s ease;
}
.acc-item.nested[open] > .acc-content {
  max-height: 1000px;
  padding: 10px 14px 14px;
  border-top: 1px dashed rgba(249, 196, 31, 0.35);
}

/* === Placeholders y listas === */
.placeholder { color: var(--amarillo-texto); font-style: italic; }
.pasos-list { margin: 0; padding-left: 1.2rem; }

/* Evitar "empujes" por márgenes colapsados del primer/último hijo */

```

```

.acc-content > *:first-child { margin-top: 0; }
.acc-content > *:last-child { margin-bottom: 0; }

/* === Responsive: tablets y móviles ===
   Ajustes por breakpoint para reducir rellenos y tipografías en pantallas
   más pequeñas. Revisa estos valores si cambias `max-width` o `min-
height`.
*/
@media (max-width: 991.98px) {
    .guia-container { padding: 32px; border-radius: 18px; min-height: 78vh;
}
    .guia-titulo { font-size: 2rem; }
}

@media (max-width: 575.98px) {
    .guia-container { padding: 20px; border-radius: 14px; max-width: 100%;
min-height: 72vh; }
    .guia-titulo { font-size: 1.6rem; letter-spacing: 1px; }
    .acc-summary { font-size: 1rem; padding: 12px 14px; }
}

/* === Scroll y barras invisibles ===
   Ocultar visualmente la barra de scroll en distintas plataformas
mientras
   se mantiene la capacidad de desplazar el contenido. Esto mejora la
estética pero comprueba accesibilidad en cada navegador.
*/
html, body {
    margin: 0;
    padding: 0;
    width: 100%;
    height: 100%;
    overflow-x: hidden;
    overflow-y: auto;
    scrollbar-width: none; /* Firefox */
    -ms-overflow-style: none;
}
html::-webkit-scrollbar, body::-webkit-scrollbar {
    width: 0px;
    background: transparent;
}

/* === Acordeón anidado: scroll interno (sin barras visibles) ===
   Cuando el acordeón anidado está abierto, limitamos la altura y
permitimos
   scroll interno ocultando visualmente la barra en la mayoría de
navegadores.
*/
.acc-item.nested[open] > .acc-content {

```

```

/* limita la altura del cajón de videos y permite scroll interno */
max-height: min(600px, 60vh); /* ajusta a tu gusto: 500px, 70vh, etc.
*/
overflow-y: auto; /* scroll vertical dentro del bloque */
overflow-x: hidden; /* sin scroll horizontal */
-webkit-overflow-scrolling: touch; /* suave en móviles */

/* oculta la barra de scroll vertical en todos los navegadores */
scrollbar-width: none; /* Firefox */
-ms-overflow-style: none; /* IE/Edge antiguo */
}

.acc-item.nested[open] > .acc-content::-webkit-scrollbar {
width: 0;
height: 0;
display: none; /* Chrome/Edge/Safari */
}

```

### | | └ Usuario.css

```

/* Usuario.css – documentación in-situ
Propósito: estilos para la sección de usuarios (tablas, formularios,
botones y tarjetas). Comentarios añadidos en español para facilitar
mantenimiento. NO se cambian reglas, sólo se documentan las secciones.

Secciones clave:
- :root -> variables de color y sombras
- Layout base (fondos decorativos)
- #usuario-container -> contenedor principal
- Formularios (#usuarioForm)
- Botones y componentes (.btn, .card, .table)
- Responsive -> media queries
*/
:root {
/* === Colores principales === */
--amarillo: #f9c41f;
--gris: #838886;
--negro: #121212;
--blanco: #fff;
--gris-fondo: #f7f7f7;

/* === Colores extendidos === */
--naranja: #ff9f1c;
--gris-claro: #f8f9fa;
--gris-muy-claro: #e9ecef;
--azul-cielo: #87ceeb;
--verde-menta: #98fb98;
--fondo-blanco: rgba(255, 255, 255, 0.95);

```

```

/* === Sistema de sombras avanzado === */
--sombra: rgba(0, 0, 0, 0.1);
--sombra-hover: rgba(0, 0, 0, 0.15);
--sombra-moderna: rgba(0, 0, 0, 0.06);
--sombra-elevada: 0 8px 32px rgba(0, 0, 0, 0.12);
--sombra-suave: 0 2px 8px rgba(0, 0, 0, 0.08);
--sombra-intensa: 0 16px 48px rgba(0, 0, 0, 0.2);
}

/* === LAYOUT BASE - Modernizado === */
/* -----
   Layout base y fondos decorativos
`body::before` y `body::after` añaden decoraciones visuales. Reducir o
eliminar estas capas en móviles si quieras mejorar rendimiento.
*/
body {
  margin: 0;
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
  display: flex;
  min-height: 100vh;
  background: var(--gris) !important;
  overflow-x: hidden;
  position: relative;
}

/* Fondos decorativos modernos con efectos avanzados */
body::before,
body::after {
  content: "";
  position: fixed;
  z-index: -1;
  pointer-events: none;
}

body::before {
  top: 0;
  left: 0;
  width: 45%;
  height: 75%;
  background: radial-gradient(circle at 25% 25%, rgba(249, 196, 31, 0.15)
0%, transparent 50%),
              linear-gradient(135deg, var(--amarillo) 0%, var(--naranja)
100%);
  border-radius: 0 0 50% 0;
  filter: blur(1px);
  animation: floatBackground 8s ease-in-out infinite;
}

```

```

body::after {
    bottom: 0;
    right: 0;
    width: 65%;
    height: 55%;
    background: radial-gradient(circle at 75% 75%, rgba(255, 159, 28, 0.12) 0%, transparent 50%),
                linear-gradient(225deg, var(--gris) 0%, #6c757d 100%);
    border-radius: 50% 0 0 0;
    filter: blur(1px);
    animation: floatBackground 10s ease-in-out infinite reverse;
}

@keyframes floatBackground {
    0%, 100% { transform: translateY(0px) scale(1); }
    50% { transform: translateY(-20px) scale(1.02); }
}

/* === CONTENEDOR PRINCIPAL - Modernizado === */
/* -----
   Contenedor principal `#usuario-container`
   Ajustar `max-width`, `padding` y `min-height` aquí para cambiar la
   densidad visual de la sección de usuarios.
*/
#usuario-container {
    max-width: 1200px;
    margin: auto;
    padding: 32px;
    background: var(--fondo-blanco);
    box-shadow: var(--sombra-elevada);
    border-radius: 20px;
    border: none;
    position: relative;
    overflow: hidden;
    backdrop-filter: blur(10px);
    transition: all 0.3s ease;
}

#usuario-container::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 6px;
    background: linear-gradient(90deg, var(--amarillo) 0%, var(--naranja) 50%, #ffd700 100%);
    z-index: 1;
}

```

```

}

#usuario-container:hover {
  transform: translateY(-4px);
  box-shadow: var(--sombra-intensa);
}

/* === FORMULARIOS - Modernizados === */
/* -----
   Formularios (`#usuarioForm`)
   Estilos reutilizables para labels, inputs, selects y textarea. Las
   reglas incluyen estados visuales para campos vacíos y focus.
*/
#usuarioForm .form-label {
  font-weight: 700;
  color: var(--negro);
  margin-bottom: 8px;
  font-size: 0.95rem;
  text-transform: uppercase;
  letter-spacing: 0.5px;
  display: block;
}

#usuarioForm input,
#usuarioForm select,
#usuarioForm textarea {
  font-size: 0.95rem;
  border-radius: 12px;
  border: 2px solid rgba(255, 255, 255, 0.3);
  padding: 14px 16px;
  transition: all 0.3s ease;
  background: rgba(255, 255, 255, 0.9);
  color: var(--negro);
  font-weight: 500;
  backdrop-filter: blur(5px);
  width: 100%;
  box-sizing: border-box;
}

/* === PLACEHOLDER Y SELECT MEJORADOS === */
#usuarioForm input::placeholder,
#usuarioForm textarea::placeholder {
  color: rgba(0, 0, 0, 0.4);
  font-weight: 400;
  font-style: italic;
}

/* Mejorando el select cuando no se ha seleccionado nada */

```

```

#usuarioForm select:invalid,
#usuarioForm select[value="""] {
  color: rgba(0, 0, 0, 0.4) !important; /* Igualado al placeholder */
  font-style: italic;
  font-weight: 400; /* Igualado al placeholder */
}

#usuarioForm select option {
  color: var(--negro) !important;
  font-style: normal;
  font-weight: 500;
}

#usuarioForm select option:first-child {
  color: rgba(0, 0, 0, 0.4) !important; /* Igualado al placeholder */
  font-style: italic;
  font-weight: 400; /* Igualado al placeholder */
}

#usuarioForm input:focus,
#usuarioForm select:focus,
#usuarioForm textarea:focus {
  border-color: var(--amarillo);
  background: rgba(255, 255, 255, 0.95);
  box-shadow: 0 0 0 4px rgba(249, 196, 31, 0.2), 0 4px 20px rgba(249, 196, 31, 0.15);
  outline: none;
  transform: translateY(-2px);
}

#usuarioForm textarea {
  resize: vertical;
  min-height: 120px;
  border-radius: 12px;
}

/* === CAMPOS VACÍOS - Más visibles === */
#usuarioForm input:not(:focus):placeholder-shown,
#usuarioForm input[type="password"]:not(:focus):not(:valid),
#usuarioForm textarea:not(:focus):placeholder-shown,
#usuarioForm textarea:not(:focus):empty,
#usuarioForm textarea:not(:focus)[value=""],
#usuarioForm select:not(:focus):invalid,
#usuarioForm select:not(:focus)[value="""] {
  background: rgba(249, 196, 31, 0.05) !important;
  border-color: rgba(249, 196, 31, 0.3) !important;
  box-shadow: inset 0 2px 4px rgba(249, 196, 31, 0.1) !important;
}

```

```

#usuarioForm input:not(:focus):placeholder-shown:hover,
#usuarioForm input[type="password"]:not(:focus):not(:valid):hover,
#usuarioForm textarea:not(:focus):placeholder-shown:hover,
#usuarioForm textarea:not(:focus):empty:hover,
#usuarioForm textarea:not(:focus)[value=""]":hover,
#usuarioForm select:not(:focus):invalid:hover,
#usuarioForm select:not(:focus)[value=""]":hover {
    background: rgba(249, 196, 31, 0.08) !important;
    border-color: rgba(249, 196, 31, 0.4) !important;
    box-shadow: 0 2px 8px rgba(249, 196, 31, 0.15) !important;
}

/* === Campos vacíos iniciales === */
#usuarioForm input:not(:focus):invalid,
#usuarioForm textarea:not(:focus):invalid {
    background: rgba(249, 196, 31, 0.05) !important;
    border-color: rgba(249, 196, 31, 0.3) !important;
    box-shadow: inset 0 2px 4px rgba(249, 196, 31, 0.1) !important;
}

#usuarioForm input:not(:focus):invalid:hover,
#usuarioForm textarea:not(:focus):invalid:hover {
    background: rgba(249, 196, 31, 0.08) !important;
    border-color: rgba(249, 196, 31, 0.4) !important;
    box-shadow: 0 2px 8px rgba(249, 196, 31, 0.15) !important;
}

.form-group {
    margin-bottom: 24px;
    position: relative;
}

/* === BOTONES - Modernizados === */
/* -----
   Botones (.btn, .btn-primary, .btn-secondary, etc.)
   Estilos del CTA y variantes. Mantén consistencia usando estas clases
   para acciones globales dentro de la UI de Usuarios.
*/
.btn {
    min-width: 140px;
    padding: 14px 28px;
    border-radius: 12px;
    font-weight: 700;
    border: none;
    cursor: pointer;
    transition: all 0.3s ease;
    position: relative;
}

```

```
overflow: hidden;
text-transform: uppercase;
letter-spacing: 0.8px;
font-size: 0.9rem;
}

.btn::before {
  content: "";
  position: absolute;
  top: 0;
  left: -100%;
  width: 100%;
  height: 100%;
  background: linear-gradient(90deg, transparent, rgba(255, 255, 255, 0.2), transparent);
  transition: left 0.5s ease;
}

.btn:hover::before {
  left: 100%;
}

.btn-primary {
  background: linear-gradient(135deg, var(--amarillo), var(--naranja));
  color: var(--negro);
  box-shadow: 0 4px 16px rgba(249, 196, 31, 0.3);
}

.btn-primary:hover {
  background: linear-gradient(135deg, #e6b619, var(--amarillo));
  transform: translateY(-3px);
  box-shadow: 0 8px 24px rgba(249, 196, 31, 0.4);
}

.btn-secondary {
  background: linear-gradient(135deg, var(--gris), #6c757d);
  color: var(--blanco);
  box-shadow: 0 4px 16px rgba(131, 136, 134, 0.3);
}

.btn-secondary:hover {
  background: linear-gradient(135deg, #6c757d, var(--gris));
  transform: translateY(-3px);
  box-shadow: 0 8px 24px rgba(131, 136, 134, 0.4);
}

.btn-success {
  background: linear-gradient(135deg, #28a745, #20c997);
```

```

color: var(--blanco);
box-shadow: 0 4px 16px rgba(40, 167, 69, 0.3);
}

.btn-success:hover {
background: linear-gradient(135deg, #218838, #28a745);
transform: translateY(-3px);
box-shadow: 0 8px 24px rgba(40, 167, 69, 0.4);
}

.btn-danger {
background: linear-gradient(135deg, #dc3545, #e55565);
color: var(--blanco);
box-shadow: 0 4px 16px rgba(220, 53, 69, 0.3);
}

.btn-danger:hover {
background: linear-gradient(135deg, #c82333, #dc3545);
transform: translateY(-3px);
box-shadow: 0 8px 24px rgba(220, 53, 69, 0.4);
}

/* === TABLAS - Modernizadas === */
/* -----
   Tablas y scroll personalizado
` .table-responsive` define un contenedor con scroll y estilo de thumb
personalizado para navegadores WebKit. En otros navegadores la barra
puede comportarse diferente.
*/
.table-responsive {
max-height: 500px;
overflow-y: auto;
border-radius: 16px;
background: var(--fondo-blanco);
box-shadow: var(--sombra-elevada);
backdrop-filter: blur(10px);
}

.table {
background: transparent;
border-radius: 16px;
overflow: hidden;
border: none;
margin: 0;
}

.table thead th {
background: linear-gradient(135deg, var(--negro), #2c2c2c);

```

```

color: var(--blanco);
font-weight: 700;
padding: 20px 16px;
border: none;
text-transform: uppercase;
letter-spacing: 0.8px;
font-size: 0.9rem;
position: sticky;
top: 0;
z-index: 10;
}

.table tbody td {
  padding: 16px;
  border: none;
  color: var(--negro);
  font-weight: 500;
  vertical-align: middle;
  background: rgba(255, 255, 255, 0.8);
  transition: all 0.3s ease;
}

.table tbody tr {
  transition: all 0.3s ease;
  border-bottom: 1px solid rgba(255, 255, 255, 0.1);
}

.table tbody tr:hover {
  transform: translateY(-2px);
  box-shadow: 0 4px 16px rgba(249, 196, 31, 0.15);
}

.table tbody tr:hover td {
  background: rgba(249, 196, 31, 0.1);
  color: var(--negro);
}

.table tbody tr:nth-child(even) td {
  background: rgba(255, 255, 255, 0.6);
}

.table tbody tr:nth-child(even):hover td {
  background: rgba(249, 196, 31, 0.12);
}

/* === SCROLL PERSONALIZADO - Modernizado === */
.table-responsive::-webkit-scrollbar {
  width: 8px;
}

```

```

    height: 8px;
}

.table-responsive::-webkit-scrollbar-track {
    background: rgba(255, 255, 255, 0.1);
    border-radius: 8px;
}

.table-responsive::-webkit-scrollbar-thumb {
    background: linear-gradient(135deg, var(--amarillo), var(--naranja));
    border-radius: 8px;
    box-shadow: 0 2px 8px rgba(249, 196, 31, 0.3);
}

.table-responsive::-webkit-scrollbar-thumb:hover {
    background: linear-gradient(135deg, #e6b619, var(--amarillo));
}

body,
html,
.container,
.content {
    scrollbar-width: none;          /* Firefox */
    -ms-overflow-style: none;       /* IE 10+ */
}

body::-webkit-scrollbar,
html::-webkit-scrollbar,
.container::-webkit-scrollbar,
.content::-webkit-scrollbar {
    display: none;                  /* Chrome, Safari, Opera */
}

/* === COMPONENTES - Modernizados === */
/* -----
   Componentes ` .card` y ` .card-header`
   Tarjetas reutilizables para agrupar contenido. La pseudo-regla ::before
   añade la franja superior dorada; cambiarla afectará todas las cards.
*/
.card {
    border: none;
    border-radius: 20px;
    overflow: hidden;
    background: var(--fondo-blanco);
    box-shadow: var(--sombra-elevada);
    transition: all 0.3s ease;
    backdrop-filter: blur(10px);
    position: relative;
}

```

```
}

.card::before {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 4px;
  background: linear-gradient(90deg, var(--amarillo) 0%, var(--naranja) 100%);
  z-index: 1;
}

.card:hover {
  transform: translateY(-6px);
  box-shadow: var(--sombra-intensa);
}

.card.shadow.mb-4 {
  background: var(--fondo-blanco);
  border: 1px solid rgba(249, 196, 31, 0.2);
  position: relative;
  overflow: hidden;
}

.card.shadow.mb-4::before {
  height: 6px;
  background: linear-gradient(90deg, var(--amarillo) 0%, var(--naranja) 50%, #ffd700 100%);
}

.card.shadow:last-of-type {
  background: var(--fondo-blanco);
  box-shadow: var(--sombra-elevada);
}

.card-header {
  background: linear-gradient(135deg, var(--negro) 0%, #2c2c2c 100%) !important;
  color: var(--blanco) !important;
  font-weight: 700;
  letter-spacing: 0.8px;
  padding: 24px 28px;
  position: relative;
  z-index: 2;
  text-transform: uppercase;
}
```

```
.card-header h5,  
.card-header h6 {  
  margin: 0;  
  font-size: 1.25rem;  
  color: var(--amarillo) !important;  
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);  
  font-weight: 700;  
}  
  
.card-body {  
  padding: 28px 32px;  
}  
  
/* === BOTONES ESPECIALIZADOS === */  
.btn-amarillo {  
  background: linear-gradient(135deg, var(--amarillo), var(--naranja)) !important;  
  border: none !important;  
  color: var(--negro) !important;  
  font-weight: 700;  
  text-transform: uppercase;  
  letter-spacing: 0.5px;  
  border-radius: 12px;  
  padding: 12px 24px;  
  box-shadow: 0 4px 16px rgba(249, 196, 31, 0.3);  
  transition: all 0.3s ease;  
  position: relative;  
  overflow: hidden;  
}  
  
.btn-amarillo::before {  
  content: "";  
  position: absolute;  
  top: 0;  
  left: -100%;  
  width: 100%;  
  height: 100%;  
  background: linear-gradient(90deg, transparent, rgba(255, 255, 255, 0.2), transparent);  
  transition: left 0.5s ease;  
}  
  
.btn-amarillo:hover::before {  
  left: 100%;  
}  
  
.btn-amarillo:hover,
```

```
.btn-amarillo:focus {
  background: linear-gradient(135deg, #e6b619, var(--amarillo))
!important;
  border: none !important;
  color: var(--negro) !important;
  transform: translateY(-3px);
  box-shadow: 0 8px 24px rgba(249, 196, 31, 0.4);
}

/* === CONTENIDO PRINCIPAL - Modernizado === */
.content {
  margin-left: 220px;
  padding: 32px;
  box-sizing: border-box;
  flex-grow: 1;
  position: relative;
  z-index: 1;
}

/* === RESPONSIVE DESIGN - Modernizado === */
@media (max-width: 992px) {
  .content {
    margin-left: 0;
    padding: 24px 20px;
    width: 100%;
    max-width: none;
  }

  #usuario-container {
    padding: 24px;
    margin: 16px;
    border-radius: 16px;
  }

  body::before {
    width: 50%;
    height: 65%;
  }

  body::after {
    width: 70%;
    height: 45%;
  }
}

@media (max-width: 768px) {
  body::before {
    width: 60%;
```

```
height: 50%;  
background: radial-gradient(circle, rgba(249, 196, 31, 0.12) 0%,  
transparent 70%);  
}  
  
body::after {  
width: 55%;  
height: 40%;  
background: radial-gradient(circle, rgba(131, 136, 134, 0.08) 0%,  
transparent 70%);  
}  
  
.content {  
padding: 16px 12px;  
width: 100%;  
max-width: none;  
margin: 0 auto;  
}  
  
#usuario-container {  
margin: 12px;  
padding: 20px;  
border-radius: 16px;  
}  
  
.card {  
margin-bottom: 20px;  
border-radius: 16px;  
box-shadow: 0 8px 32px rgba(0, 0, 0, 0.15);  
}  
  
.card-header {  
text-align: center;  
font-size: 1.1rem;  
padding: 16px 20px;  
}  
  
.btn {  
width: 100%;  
margin-top: 12px;  
padding: 12px 20px;  
}  
  
.d-flex.justify-content-end {  
flex-direction: column;  
align-items: stretch;  
}
```

```
.table-responsive {  
    width: 100%;  
    overflow-x: auto;  
    border-radius: 12px;  
}  
  
.table-responsive table {  
    width: 100%;  
    table-layout: auto;  
}  
}  
  
}  
  
@media (max-width: 576px) {  
    body::before,  
    body::after {  
        display: none;  
    }  
  
.content {  
    padding: 12px 8px;  
    width: 100%;  
    max-width: 600px;  
    margin: 0 auto;  
    box-sizing: border-box;  
}  
  
}  
  
#usuario-container {  
    margin: 8px;  
    padding: 16px;  
    border-radius: 12px;  
}  
  
}  
  
.card {  
    width: 100%;  
    margin-bottom: 16px;  
    border-radius: 12px;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.08);  
}  
  
}  
  
.card-header {  
    padding: 12px 16px;  
    font-size: 1rem;  
}  
  
}  
  
.btn {  
    width: 100%;  
    padding: 10px 16px;  
    font-size: 0.9rem;  
}
```

```
    margin-top: 10px;
}

#usuarioForm input,
#usuarioForm select,
#usuarioForm textarea {
    padding: 12px 14px;
    font-size: 0.9rem;
    border-radius: 10px;
}

.usuarios-header {
    flex-direction: column;
    align-items: stretch;
    gap: 12px;
    padding: 16px 20px;
}

.buscador-usuarios {
    max-width: 100%;
    min-width: 0;
}

}

@media (max-width: 480px) {
    .content {
        padding: 8px 4px;
    }

    #usuario-container {
        margin: 4px;
        padding: 12px;
    }

    .card-header {
        padding: 10px 12px;
    }

    .btn {
        padding: 8px 12px;
        font-size: 0.85rem;
    }
}

/* === ELEMENTOS ESPECIALES - Modernizados === */
.mover-derecha {
    max-width: 100%;
    transform: none;
}
```

```
    margin: 0 auto;
}

.usuarios-header {
  background: linear-gradient(135deg, var(--negro) 0%, #2c2c2c 100%);
  border-radius: 16px 16px 0 0;
  box-shadow: 0 4px 16px rgba(0, 0, 0, 0.1);
  padding: 20px 24px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.titulo-usuarios {
  color: var(--amarillo) !important;
  font-weight: 700;
  margin-bottom: 0;
  text-transform: uppercase;
  letter-spacing: 0.8px;
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);
  font-size: 1.2rem;
}

.buscador-usuarios {
  max-width: 320px;
  min-width: 240px;
  font-weight: 500;
  border-radius: 25px;
  background: rgba(255, 255, 255, 0.95);
  color: var(--negro);
  box-shadow: 0 4px 16px rgba(0, 0, 0, 0.15);
  border: 2px solid rgba(255, 255, 255, 0.3);
  padding: 12px 20px;
  backdrop-filter: blur(10px);
  transition: all 0.3s ease;
}

.buscador-usuarios::placeholder {
  color: rgba(0, 0, 0, 0.5);
  font-weight: 400;
}

.buscador-usuarios:focus {
  border-color: var(--amarillo);
  box-shadow: 0 0 0 3px rgba(249, 196, 31, 0.2), 0 4px 20px rgba(249, 196, 31, 0.15);
  outline: none;
  transform: translateY(-2px);
```

```

}

/* === EFECTOS ESPECIALES === */
.card:hover .card-header {
  background: linear-gradient(135deg, #2c2c2c 0%, var(--negro) 100%);
}

.table tbody tr:hover .btn {
  transform: scale(1.05);
}

/* === TRANSICIONES GLOBALES === */
* {
  transition: all 0.3s ease;
}

/* Estilos para el highlight del usuario recién guardado */
.usuario-destacado {
  background-color: #d4edda !important;
  border-left: 4px solid #28a745 !important;
  box-shadow: 0 2px 8px rgba(40, 167, 69, 0.3) !important;
  animation: pulseHighlight 2s ease-in-out;
}

@keyframes pulseHighlight {
  0% {
    background-color: #d4edda;
    transform: scale(1);
  }
  50% {
    background-color: #c3e6cb;
    transform: scale(1.02);
  }
  100% {
    background-color: #d4edda;
    transform: scale(1);
  }
}

/* Mejorar la interactividad visual de las filas clickeables */
.fila-usuario {
  transition: all 0.2s ease-in-out;
}

.fila-usuario:hover {
  background-color: #f8f9fa !important;
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  cursor: pointer;
}

```

```

}

/* Indicador visual de que la fila es clickeable */
.fila-usuario td:first-child::before {
    content: "✎";
    opacity: 0;
    transition: opacity 0.2s ease-in-out;
    margin-right: 8px;
}

.fila-usuario:hover td:first-child::before {
    opacity: 0.5;
}

/* Mejorar la apariencia de los botones en la tabla */
.fila-usuario .btn {
    transition: all 0.2s ease-in-out;
}

.fila-usuario .btn:hover {
    transform: translateY(-1px);
    box-shadow: 0 2px 4px rgba(0,0,0,0.2);
}

/* === FIN DE MODERNIZACIÓN USUARIO.CSS === */

```

|   |   └── img/  
|   |   └── Amarillo\_LBlancas.png



B A S T O S  
R E S T A U R A N T E

|   |   └── Amarillo\_LNegras.png



B A S T O S

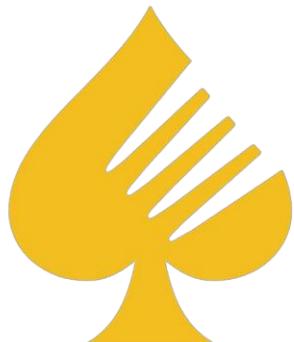
| | └─ Negro\_LNegras.png



B A S T O S

RESTAURANTE

| | └─ Principal\_Amarillo.png



| | └─ Principal\_Negro.png



| └─ js/  
| └─ Analysis.js

```

// public/js/Analisis.js

/**
 * Script de análisis de clientes y ventas.
 *
 * Funcionalidades principales:
 * - Ranking de clientes frecuentes, inactivos, antiguos y con mayor
historial.
 * - Ranking de residencias frecuentes.
 * - Estadísticas de ventas por mes y año.
 * - Gráficos dinámicos con Chart.js.
 * - Buscador en vivo adaptado a cada tipo de análisis.
 */

// Guarda el ranking actual para filtrar en vivo
let datosAnalisisActual = [];
// Si estamos en modo residencias frecuentes (busca por residencia)
let modoResidencia = false;
// Guarda la función del análisis actual para el botón "Actualizar"
let lastAnalisisFuncion = mostrarClientesFrecuentes;

document.addEventListener("DOMContentLoaded", () => {
    /**
     * === Asignación de eventos a los botones de análisis ===
     */
    document
        .getElementById("btnClientesFrecuentes")
        .addEventListener("click", () => {
            lastAnalisisFuncion = mostrarClientesFrecuentes;
            mostrarClientesFrecuentes();
        });
    document
        .getElementById("btnClientesMayorHistorial")
        .addEventListener("click", () => {
            lastAnalisisFuncion = mostrarClientesMayorHistorial;
            mostrarClientesMayorHistorial();
        });
    document
        .getElementById("btnClientesInactivos")
        .addEventListener("click", () => {
            lastAnalisisFuncion = mostrarClientesInactivos;
            mostrarClientesInactivos();
        });
    document
        .getElementById("btnResidenciasFrecuentes")
        .addEventListener("click", () => {
            lastAnalisisFuncion = mostrarResidenciasFrecuentes;
            mostrarResidenciasFrecuentes();
        });
}

```

```

    });
document
    .getElementById("btnClientesAntiguos")
    .addEventListener("click", () => {
        lastAnalisisFuncion = mostrarClientesAntiguos;
        mostrarClientesAntiguos();
    });

document
    .getElementById("btnVentasPorMes")
    .addEventListener("click", () => {
        lastAnalisisFuncion = mostrarVentasPorMes;
        mostrarVentasPorMes();
    });

document
    .getElementById("btnVentasPorAnio")
    .addEventListener("click", () => {
        lastAnalisisFuncion = mostrarVentasPorAnio;
        mostrarVentasPorAnio();
    });

// Botón de actualizar recarga el análisis actual
document
    .getElementById("btnActualizarAnalisis")
    .addEventListener("click", () => {
        if (lastAnalisisFuncion) lastAnalisisFuncion();
    });

/**
 * === Buscador general dinámico ===
 * Filtra la tabla dependiendo del análisis en curso
 */
document
    .getElementById("analisisBuscadorGeneral")
    .addEventListener("input", function () {
        const value = this.value.trim().toLowerCase();
        let filtrados = [];

        // --- Residencias frecuentes (busca por lugar de residencia) ---
        if (modoResidencia) {
            filtrados = !value
                ? datosAnalisisActual
                : datosAnalisisActual.filter(r =>
                    (r.residencia ?? "").toLowerCase().includes(value)
                );
            renderResidenciasFrecuentesTabla(filtrados);
        }
    });

```

```

// --- Ventas por mes ---
} else if (lastAnalisisFuncion === mostrarVentasPorMes) {
    filtrados = !value
    ? datosAnalisisActual
    : datosAnalisisActual.filter(
        v =>
        (v.mes ?? "").toLowerCase().includes(value) ||
        (v.año ?? "").toString().includes(value)
    );
    renderTablaVentasPorMes(filtrados);

// --- Ventas por año ---
} else if (lastAnalisisFuncion === mostrarVentasPorAnio) {
    filtrados = !value
    ? datosAnalisisActual
    : datosAnalisisActual.filter(
        v => (v.año ?? "").toString().includes(value)
    );
    renderTablaVentasPorAnio(filtrados);

// --- Otros análisis (buscan por nombre) ---
} else {
    filtrados = !value
    ? datosAnalisisActual
    : datosAnalisisActual.filter(
        c => (c.nombre ?? "").toLowerCase().includes(value)
    );
    renderTablaAnalisisActual(filtrados);
}

/**
 * === Detectar parámetro URL para auto-seleccionar sección ===
 */
const urlParams = new URLSearchParams(window.location.search);
const seccion = urlParams.get('seccion');

// Cargar sección específica o la por defecto
switch(seccion) {
    case 'ventasPorMes':
        console.log('Auto-cargando: Ventas por mes');
        lastAnalisisFuncion = mostrarVentasPorMes;
        mostrarVentasPorMes();
        break;
    case 'ventasPorAnio':
        console.log('Auto-cargando: Ventas por año');
        lastAnalisisFuncion = mostrarVentasPorAnio;
        mostrarVentasPorAnio();
}

```

```

        break;
    case 'clientesFrecuentes':
        console.log('Auto-cargando: Clientes frecuentes');
        lastAnalisisFuncion = mostrarClientesFrecuentes;
        mostrarClientesFrecuentes();
        break;
    case 'clientesMayorHistorial':
        console.log('Auto-cargando: Clientes con mayor historial');
        lastAnalisisFuncion = mostrarClientesMayorHistorial;
        mostrarClientesMayorHistorial();
        break;
    case 'clientesInactivos':
        console.log('Auto-cargando: Clientes inactivos');
        lastAnalisisFuncion = mostrarClientesInactivos;
        mostrarClientesInactivos();
        break;
    case 'residenciasFrecuentes':
        console.log('Auto-cargando: Residencias frecuentes');
        lastAnalisisFuncion = mostrarResidenciasFrecuentes;
        mostrarResidenciasFrecuentes();
        break;
    case 'clientesAntiguos':
        console.log('Auto-cargando: Clientes antiguos');
        lastAnalisisFuncion = mostrarClientesAntiguos;
        mostrarClientesAntiguos();
        break;
    default:
        // Por defecto, carga el primer análisis al entrar
        mostrarClientesFrecuentes();
        break;
    }
});

/***
 * === ANÁLISIS 1: CLIENTES MÁS FRECUENTES ===
 * Llama al backend y muestra clientes con más visitas
 */
async function mostrarClientesFrecuentes() {
    setBotonActivo("btnClientesFrecuentes");
    modoResidencia = false;

    // Limpiar y mostrar loader
    setBuscador("Buscar por nombre...");
    const tablaCont = document.getElementById("analisisTablaCont");
    const graficoCont = document.getElementById("analisisGraficoCont");
    tablaCont.innerHTML = loaderHTML();
    graficoCont.innerHTML = "";
}

```

```

try {
    const res = await
fetch("/CRM_INT/CRM/controller/AnalisisController.php", {
    method: "POST",
    headers: { "Content-Type": "application/x-www-form-urlencoded" },
    body: "action=clientesFrecuentes",
});
const json = await res.json();
if (!json.success) throw new Error(json.message);

const ranking = json.data.ranking;
datosAnalisisActual = ranking; // Actualiza ranking para filtro
const top = json.data.top;

renderTablaAnalisisActual(ranking);

// Gráfico Top 5
renderGraficoBarras({
    cont: graficoCont,
    idCanvas: "graficoFrecuentes",
    labels: top.map((c) =>
        c.nombre.length > 15 ? c.nombre.slice(0, 15) + "..." : c.nombre
    ),
    data: top.map((c) => c.visitas),
    label: "Visitas",
    color: "var(--amarillo)",
    tooltipTitle: (i) => top[i].nombre,
    tooltipLabel: (i) => [
        `Visitas: ${top[i].visitas}`,
        `Tarjeta: ${top[i].id}`,
    ],
});
} catch (e) {
    tablaCont.innerHTML = errorHTML(e.message);
    graficoCont.innerHTML = "";
}
}

/**
 * === ANÁLISIS 2: CLIENTES CON MAYOR HISTORIAL DE COMPRAS ===
 */
async function mostrarClientesMayorHistorial() {
    setBotonActivo("btnClientesMayorHistorial");
    modoResidencia = false;
    setBuscador("Buscar por nombre...");
    const tablaCont = document.getElementById("analisisTablaCont");
    const graficoCont = document.getElementById("analisisGraficoCont");
    tablaCont.innerHTML = loaderHTML();
}

```

```

graficoCont.innerHTML = "";

try {
  const res = await
fetch("/CRM_INT/CRM/controller/AnalisisController.php", {
  method: "POST",
  headers: { "Content-Type": "application/x-www-form-urlencoded" },
  body: "action=clientesMayorHistorial",
});
const json = await res.json();
if (!json.success) throw new Error(json.message);

const ranking = json.data.ranking;
datosAnalisisActual = ranking;
const top = json.data.top;

renderTablaAnalisisActual(ranking);

renderGraficoBarras({
  cont: graficoCont,
  idCanvas: "graficoMayorHistorial",
  labels: top.map((c) =>
    c.nombre.length > 15 ? c.nombre.slice(0, 15) + "..." : c.nombre
  ),
  data: top.map((c) => c.totalHistorico),
  label: "Total histórico",
  color: "var(--amarillo)",
  tooltipTitle: (i) => top[i].nombre,
  tooltipLabel: (i) => [
    `Total histórico:
${parseFloat(top[i].totalHistorico).toLocaleString(
      "es-CR"
)} `,
    `Tarjeta: ${top[i].id}`,
  ],
});
} catch (e) {
  tablaCont.innerHTML = errorHTML(e.message);
  graficoCont.innerHTML = "";
}

/***
 * === ANÁLISIS 3: CLIENTES INACTIVOS ===
 * Muestra clientes que no han comprado en X días
 */
async function mostrarClientesInactivos() {

```

```

setBotonActivo("btnClientesInactivos");
modoResidencia = false;
setBuscador("Buscar por nombre...");
const tablaCont = document.getElementById("analisisTablaCont");
const graficoCont = document.getElementById("analisisGraficoCont");
tablaCont.innerHTML = loaderHTML();
graficoCont.innerHTML = "";

try {
  const res = await
fetch("/CRM_INT/CRM/controller/AnalisisController.php", {
  method: "POST",
  headers: { "Content-Type": "application/x-www-form-urlencoded" },
  body: "action=clientesInactivos",
});
const json = await res.json();
if (!json.success) throw new Error(json.message);

const ranking = json.data.ranking;
datosAnalisisActual = ranking;
const top = json.data.top;
const dias = json.diasInactivo;

if (!ranking.length) {
  tablaCont.innerHTML = `<div class="alert alert-success mt-4">Aún no
hay clientes que lleven más de ${dias} días sin comprar.</div>`;
  graficoCont.innerHTML = "";
  return;
}

// Render tabla con info de días sin comprar
renderTablaAnalisisActual(ranking);

renderGraficoBarras({
  cont: graficoCont,
  idCanvas: "graficoInactivos",
  labels: top.map((c) =>
    c.nombre.length > 15 ? c.nombre.slice(0, 15) + "..." : c.nombre
  ),
  data: top.map((c) => c.diasSinComprar),
  label: "Tiempo sin comprar (días)",
  color: "var(--amarillo)",
  tooltipTitle: (i) => top[i].nombre,
  tooltipLabel: (i) => [
    `Tiempo sin comprar:
${formatearAntiguedad(top[i].diasSinComprar)}`,
    `Tarjeta: ${top[i].id}`,
  ],
}

```

```

    });
} catch (e) {
    tablaCont.innerHTML = errorHTML(e.message);
    graficoCont.innerHTML = "";
}
}

/***
 * === ANÁLISIS 4: RESIDENCIAS MÁS FRECUENTES ===
 */
async function mostrarResidenciasFrecuentes() {
    setBotonActivo("btnResidenciasFrecuentes");
    modoResidencia = true;
    setBuscador("Buscar por lugar de residencia...");
    const tablaCont = document.getElementById("analisisTablaCont");
    const graficoCont = document.getElementById("analisisGraficoCont");
    tablaCont.innerHTML = loaderHTML();
    graficoCont.innerHTML = "";

    try {
        const res = await
fetch("/CRM_INT/CRM/controller/AnalisisController.php", {
            method: "POST",
            headers: { "Content-Type": "application/x-www-form-urlencoded" },
            body: "action=residenciasFrecuentes",
        });
        const json = await res.json();
        if (!json.success) throw new Error(json.message);

        const ranking = json.data.ranking;
        datosAnalisisActual = ranking;
        const top = json.data.top;

        renderResidenciasFrecuentesTabla(ranking);

        renderGraficoBarras({
            cont: graficoCont,
            idCanvas: "graficoResidencias",
            labels: top.map((r) =>
                r.residencia.length > 15
                    ? r.residencia.slice(0, 15) + "..."
                    : r.residencia
            ),
            data: top.map((r) => r.cantidad),
            label: "Clientes",
            color: "var(--amarillo)",
            tooltipTitle: (i) => top[i].residencia,
            tooltipLabel: (i) => [`Clientes: ${top[i].cantidad}`],
        });
    }
}

```

```

    });
} catch (e) {
    tablaCont.innerHTML = errorHTML(e.message);
    graficoCont.innerHTML = "";
}
}

/***
 * === ANÁLISIS 5: CLIENTES MÁS ANTIGUOS ===
 */
async function mostrarClientesAntiguos() {
    setBotonActivo("btnClientesAntiguos");
    modoResidencia = false;
    setBuscador("Buscar por nombre...");
    const tablaCont = document.getElementById("analisisTablaCont");
    const graficoCont = document.getElementById("analisisGraficoCont");
    tablaCont.innerHTML = loaderHTML();
    graficoCont.innerHTML = "";

    try {
        const res = await
fetch("/CRM_INT/CRM/controller/AnalisisController.php", {
            method: "POST",
            headers: { "Content-Type": "application/x-www-form-urlencoded" },
            body: "action=clientesAntiguos",
        });
        const json = await res.json();
        if (!json.success) throw new Error(json.message);

        const ranking = json.data.ranking;
        datosAnalisisActual = ranking;
        const top = json.data.top;

        renderTablaAnalisisActual(ranking);

        renderGraficoBarras({
            cont: graficoCont,
            idCanvas: "graficoAntiguos",
            labels: top.map((c) =>
                c.nombre.length > 15 ? c.nombre.slice(0, 15) + "..." : c.nombre
            ),
            data: top.map((c) => c.antiguedadDias),
            label: "Antigüedad (días)",
            color: "var(--amarillo)",
            indexAxis: "y",
            tooltipTitle: (i) => top[i].nombre,
            tooltipLabel: (i) => [
                `Antigüedad: ${formatarAntiguedad(top[i].antiguedadDias)}`
            ]
        });
    }
}

```

```

        `Tarjeta: ${top[i].id}`,
    ],
});
} catch (e) {
    tablaCont.innerHTML = errorHTML(e.message);
    graficoCont.innerHTML = "";
}
}

/***
 * ===== FUNCIONES DE AYUDA / UI =====
 */

// Pone el botón de análisis activo con color
function setBotonActivo(idBtn) {
    document
        .querySelectorAll(".btn-analisis")
        .forEach((b) => b.classList.remove("active"));
    document.getElementById(idBtn).classList.add("active");
}

// Cambia el placeholder y limpia el input de búsqueda
function setBuscador(texto) {
    const buscador = document.getElementById("analisisBuscadorGeneral");
    if (buscador) {
        buscador.value = "";
        buscador.placeholder = texto;
    }
}

// Loader estándar
function loaderHTML() {
    return `<div class="text-center py-4"><div class="spinner-border"
role="status"></div></div>`;
}

// Mensaje de error estándar
function errorHTML(msg) {
    return `<div class="alert alert-danger">Error: ${msg}</div>`;
}

// Utilidad para convertir días a formato legible (años, meses, semanas,
// días)
function formatearAntiguedad(dias) {
    const años = Math.floor(dias / 365);
    const meses = Math.floor((dias % 365) / 30.44);
    const semanas = Math.floor(((dias % 365) % 30.44) / 7);
    const diasRestantes = Math.round(((dias % 365) % 30.44) % 7);
}

```

```

let partes = [];
if (años) partes.push(años === 1 ? "1 año" : `${años} años`);
if (meses) partes.push(meses === 1 ? "1 mes" : `${meses} meses`);
if (semanas) partes.push(semanas === 1 ? "1 semana" : `${semanas} semanas`);
if (diasRestantes)
    partes.push(diasRestantes === 1 ? "1 día" : `${diasRestantes} días`);
return partes.length ? partes.join(", ") : "Hoy";
}

/***
 * Redibuja la tabla según el análisis activo y el array filtrado
 */
function renderTablaAnalisisActual(arr) {
    // Detecta el análisis activo por el botón con .active
    const idActivo = document.querySelector(".btn-analisis.active")?.id;
    const tablaCont = document.getElementById("analisisTablaCont");
    if (!idActivo || !tablaCont) return;

    switch (idActivo) {
        case "btnClientesFrecuentes":
            tablaCont.innerHTML =
                `<table class="table table-striped table-hover align-middle mb-0 rounded">
                    <thead class="table-dark">
                        <tr>
                            <th>#</th>
                            <th>Nombre</th>
                            <th>Cédula</th>
                            <th>Teléfono</th>
                            <th>Visitas</th>
                            <th>Última visita</th>
                        </tr>
                    </thead>
                    <tbody>
${arr
    .map(
        (cliente) => `
            <tr>
                <td>${cliente.posicion}</td>
                <td>${cliente.nombre}</td>
                <td>${cliente.cedula}</td>
                <td>${cliente.telefono ?? "-"}</td>
                <td>${cliente.visitas}</td>
                <td>${cliente.ultimaVisita ?? "-"}</td>
            </tr>
        `
    )
}
    )
}

```

```

        .join(""))
    </tbody>
</table>
`;
break;
case "btnClientesMayorHistorial":
    tablaCont.innerHTML =
        `<table class="table table-striped table-hover align-middle mb-0
rounded">
            <thead class="table-dark">
                <tr>
                    <th>#</th>
                    <th>Nombre</th>
                    <th>Cédula</th>
                    <th>Teléfono</th>
                    <th>Total Histórico</th>
                    <th>Última compra</th>
                </tr>
            </thead>
            <tbody>
                ${arr
                    .map(
                        (cliente) =>
                            <tr>
                                <td>${cliente.posicion}</td>
                                <td>${cliente.nombre}</td>
                                <td>${cliente.cedula}</td>
                                <td>${cliente.telefono ?? "-"}</td>
                                <td>${parseFloat(cliente.totalHistorico).toLocaleString(
                                    "es-CR"
                                )}</td>
                                <td>${cliente.ultimaCompra ?? "-"}</td>
                            </tr>
                        )
                    .join("")
                }
            </tbody>
        </table>
`;
break;
case "btnClientesInactivos":
    tablaCont.innerHTML =
        `<div style="max-height:370px;overflow:auto">
            <table class="table table-striped table-hover align-middle mb-0
rounded">
                <thead class="table-dark">
                    <tr>
```

```

<th>#</th>
<th>Nombre</th>
<th>Cédula</th>
<th>Teléfono</th>
<th>Última compra</th>
<th>Tiempo sin comprar</th>
<th>Total histórico</th>
</tr>
</thead>
<tbody>
${arr
.map(
(c, i) => `

<tr>
<td>${i + 1}</td>
<td>${c.nombre}</td>
<td>${c.cedula}</td>
<td>${c.telefono ?? "-"}</td>
<td>${c.ultimaCompra ?? "-"}</td>
<td>${formatearAntiguedad(c.diasSinComprar)}</td>
<td>${parseFloat(c.totalHistorico).toLocaleString(
    "es-CR"
)}</td>
</tr>
`)

)
.join(""))
</tbody>
</table>
</div>
`;
break;
case "btnClientesAntiguos":
tablaCont.innerHTML = `





```

```

        .map(
            (cliente) => `
                <tr>
                    <td>${cliente.posicion}</td>
                    <td>${cliente.nombre}</td>
                    <td>${cliente.cedula}</td>
                    <td>${formatearAntiguedad(cliente.antiguedadDias)}</t
d>
                    <td>${formatearAntiguedad(cliente.diasSinComprar)}</t
d>
                    <td>${parseFloat(cliente.totalHistorico).toLocaleStr
ing(
                        "es-CR"
                    )}</td>
                </tr>
            `
        )
        .join("")
    </tbody>
</table>
`;
break;
}
}

/**
 * Dibuja la tabla de residencias frecuentes (con collapses)
 */
function renderResidenciasFrecuentesTabla(arr) {
    const tablaCont = document.getElementById("analisisTablaCont");
    if (!tablaCont) return;
    if (!arr.length) {
        tablaCont.innerHTML = `<div class="alert alert-info mt-4">No hay
datos de residencia disponibles.</div>`;
        return;
    }
    let filas = arr
        .map((r, i) => {
            const collapseId = "collapseRes" + i;
            return `
                <tr>
                    <td>${i + 1}</td>
                    <td>${r.residencia}</td>
                    <td>${r.cantidad}</td>
                    <td>
                        <button class="btn btn-sm btn-outline-primary"
                            type="button"
                            data-bs-toggle="collapse"
                            data-bs-target="#collapseRes${i}"
                        >Ver más</button>
                    </td>
                </tr>
            `;
        })
        .join("");
    tablaCont.innerHTML = `



`;
}

```

```

        data-bs-target="#${collapseId}"
        aria-expanded="false" aria-controls="${collapseId}">
    Ver nombres
  </button>
</td>
</tr>
<tr>
  <td colspan="4" class="bg-light p-0">
    <div class="collapse" id="${collapseId}">
      <div style="max-height:150px;overflow:auto; padding:
1rem;">
        <ul class="mb-0 small">
          ${r.clientes
            .map(
              (c) => `
                <li><strong>${c.nombre}</strong> <span
class="text-muted">-- Tarjeta: ${c.id}</span></li>
              `
            )
            .join("")
          )
        </ul>
      </div>
    </div>
  </td>
</tr>
`;;
})
.join("");
}

tablaCont.innerHTML = `

<div style="max-height:370px;overflow:auto">
  <table class="table table-striped table-hover align-middle mb-0
rounded">
    <thead class="table-dark">
      <tr>
        <th>#</th>
        <th>Lugar de residencia</th>
        <th>Clientes</th>
        <th></th>
      </tr>
    </thead>
    <tbody>${filas}</tbody>
  </table>
</div>
`;
}
```

```

```

* Gráfico de barras usando Chart.js
*/
function renderGraficoBarras({
  cont,
  idCanvas,
  labels,
  data,
  label,
  color,
  tooltipTitle,
  tooltipLabel,
  indexAxis,
}) {
  cont.innerHTML = `<canvas id="${idCanvas}"></canvas>`;
  const ctx = document.getElementById(idCanvas).getContext("2d");
  new Chart(ctx, {
    type: "bar",
    data: {
      labels,
      datasets: [
        {
          label,
          data,
          backgroundColor: color,
          borderRadius: 10,
        },
      ],
    },
    options: {
      indexAxis: indexAxis || "x",
      plugins: {
        tooltip: {
          callbacks: {
            title: function (context) {
              return tooltipTitle(context[0].dataIndex);
            },
            label: function (context) {
              return tooltipLabel(context.dataIndex);
            },
          },
        },
      },
    },
  });
}

/**
 * Redibuja la tabla de ventas por mes

```

```

        */
function renderTablaVentasPorMes(arr) {
    const tablaCont = document.getElementById("analisisTablaCont");
    let filas = arr.map((v, i) => {
        let varHtml = "-";
        if (typeof v.variacion === "number") {
            varHtml = v.variacion > 0
                ? `<span class="text-success fw-bold">+${v.variacion}% <i class="fa fa-arrow-up"></i></span>`
                : `<span class="text-danger fw-bold">${v.variacion}% <i class="fa fa-arrow-down"></i></span>`;
        }
        return `
            <tr>
                <td>${v.posicion}</td>
                <td>${v.mes}</td>
                <td>${v.año}</td>
                <td>${parseFloat(v.total).toLocaleString("es-CR")}</td>
                <td>${varHtml}</td>
            </tr>
        `;
    }).join("");
}

tablaCont.innerHTML = `
    <table class="table table-striped table-hover align-middle mb-0 rounded">
        <thead class="table-dark">
            <tr>
                <th>#</th>
                <th>Mes</th>
                <th>Año</th>
                <th>Ventas Totales</th>
                <th>Var. Mensual (%)</th>
            </tr>
        </thead>
        <tbody>${filas}</tbody>
    </table>
`;
}

/**
 * Muestra análisis de ventas por mes
 */
async function mostrarVentasPorMes() {
    setBotonActivo("btnVentasPorMes");
    modoResidencia = false;
    setBuscador("Buscar por mes o año...");
    const tablaCont = document.getElementById("analisisTablaCont");

```

```

const graficoCont = document.getElementById("analisisGraficoCont");
tablaCont.innerHTML = loaderHTML();
graficoCont.innerHTML = "";

try {
    const res = await
fetch("/CRM_INT/CRM/controller/AnalisisController.php", {
    method: "POST",
    headers: { "Content-Type": "application/x-www-form-urlencoded" },
    body: "action=ventasPorMes",
});
const json = await res.json();
if (!json.success) throw new Error(json.message);

const ventas = json.data.ventas;
datosAnalisisActual = ventas;

renderTablaVentasPorMes(ventas);

// Gráfico
if (ventas.length) {
    const labels = ventas.map(v => `${v.mes} ${v.año}`).reverse();
    const data = ventas.map(v => v.total).reverse();
    graficoCont.innerHTML = `<canvas id="graficoVentasMes"></canvas>`;
    const ctx =
document.getElementById("graficoVentasMes").getContext("2d");
    new Chart(ctx, {
        type: "bar",
        data: {
            labels,
            datasets: [
                {
                    label: "Ventas Totales",
                    data,
                    backgroundColor: "var(--amarillo)",
                    borderRadius: 10,
                }],
        },
        options: {
            plugins: {
                tooltip: {
                    callbacks: {
                        title: function(context) {
                            return labels[context[0].dataIndex];
                        },
                        label: function(context) {
                            return
`#${parseFloat(data[context.dataIndex]).toLocaleString("es-CR")}`;
                        },
                    }
                }
            }
        }
    });
}
}

```

```

        },
        },
        },
        });
    }
} catch (e) {
    tablaCont.innerHTML = errorHTML(e.message);
    graficoCont.innerHTML = "";
}
}

/**
 * Muestra análisis de ventas por año
 */
async function mostrarVentasPorAnio() {
    setBotonActivo("btnVentasPorAnio");
    modoResidencia = false;
    setBuscador("Buscar por año...");
    const tablaCont = document.getElementById("analisisTablaCont");
    const graficoCont = document.getElementById("analisisGraficoCont");
    tablaCont.innerHTML = loaderHTML();
    graficoCont.innerHTML = "";

    try {
        const res = await
fetch("/CRM_INT/CRM/controller/AnalisisController.php", {
            method: "POST",
            headers: { "Content-Type": "application/x-www-form-urlencoded" },
            body: "action=ventasPorAnio",
        });
        const json = await res.json();
        if (!json.success) throw new Error(json.message);

        const ventas = json.data.ventas;
        datosAnalisisActual = ventas;

        renderTablaVentasPorAnio(ventas);

        // Gráfico de barras (años en orden ascendente)
        if (ventas.length) {
            const labels = ventas.map(v => v.año).reverse();
            const data = ventas.map(v => v.total).reverse();
            graficoCont.innerHTML = `<canvas id="graficoVentasAnio"></canvas>`;
            const ctx =
document.getElementById("graficoVentasAnio").getContext("2d");
            new Chart(ctx, {
                type: "bar",

```

```

        data: {
            labels,
            datasets: [{

                label: "Ventas Totales",
                data,
                backgroundColor: "var(--amarillo)",
                borderRadius: 10,
            }],
        },
        options: {
            plugins: {
                tooltip: {
                    callbacks: {
                        title: function(context) {
                            return `Año ${labels[context[0].dataIndex]}`;
                        },
                        label: function(context) {
                            return
`parseFloat(data[context.dataIndex]).toLocaleString("es-CR")`;
                        },
                    },
                },
            },
        });
    }
} catch (e) {
    tablaCont.innerHTML = errorHTML(e.message);
    graficoCont.innerHTML = "";
}
}

/**
 * Redibuja la tabla de ventas por año
 */
function renderTablaVentasPorAnio(arr) {
    const tablaCont = document.getElementById("analisisTablaCont");
    let filas = arr.map((v, i) => {
        let varHtml = "-";
        if (typeof v.variacion === "number") {
            varHtml = v.variacion > 0
            ? `<span class="text-success fw-bold">+${v.variacion}% <i
class="fa fa-arrow-up"></i></span>`
            : `<span class="text-danger fw-bold">${v.variacion}% <i class="fa
fa-arrow-down"></i></span>`;
        }
        return `
<tr>

```

```

        <td>${v.posicion}</td>
        <td>${v.año}</td>
        <td>${parseFloat(v.total).toLocaleString("es-CR")}</td>
        <td>${varHtml}</td>
    </tr>
    `;
}).join("");
}

tablaCont.innerHTML = `





```

## |   | — Bitacora.js

```

/**
 * =====
 * Gestión de Bitácora de Usuarios
 * =====
 *
 * Funciones principales:
 * - Cargar la bitácora de usuarios desde el backend.
 * - Mostrar los registros en tabla HTML.
 * - Calcular duración de sesiones.
 * - Exportar la tabla a PDF y Excel.
 */

/**
 * Espera a que el contenedor de la tabla esté disponible en el DOM
 * antes de cargar los datos de la bitácora.
 */
function esperarContenedorYcargar() {
    const contenedor = document.getElementById("contenedorTabla");

    if (!contenedor) {
        console.warn("⚠ Esperando a que cargue #contenedorTabla...");
    }
}

```

```

        return setTimeout(esperarContenedorYcargar, 100);
    }

    console.log("📦 contenedorTabla encontrado, cargando bitácora...");
    cargarBitacora();
}

// Ejecutar cuando se cargue el DOM
document.addEventListener("DOMContentLoaded", esperarContenedorYcargar);

/***
 * Carga los registros de bitácora desde el backend,
 * construye la tabla y la inserta en el contenedor.
 */
async function cargarBitacora() {
    try {
        const res = await fetch(
            `http://${location.hostname}/CRM_INT/CRM/controller/BitacoraController.php?action=readAll`
        );

        const text = await res.text();
        console.log("📦 Respuesta cruda:", text);
        const json = JSON.parse(text);

        const contenedor = document.getElementById("contenedorTabla");
        if (!contenedor) return;

        if (!json.success || !json.data || json.data.length === 0) {
            contenedor.innerHTML = "<p>No hay datos para mostrar.</p>";
            return;
        }

        /**
         * Calcula la duración de la sesión a partir de la hora de entrada y
         * salida.
         * @param {string} horaEntrada - Hora de entrada en formato HH:mm:ss
         * @param {string} horaSalida - Hora de salida en formato HH:mm:ss
         * @returns {string} Duración en formato `Xh Xm Xs` o `EN CURSO`
         */
        function calcularDuracion(horaEntrada, horaSalida) {
            if (!horaSalida || horaSalida === "00:00:00") return "EN CURSO";

            const [hEntH, hEntM, hEntS] = horaEntrada.split(":").map(Number);
            const [hSalH, hSalM, hSalS] = horaSalida.split(":").map(Number);

            const entrada = new Date(0, 0, 0, hEntH, hEntM, hEntS);
            const salida = new Date(0, 0, 0, hSalH, hSalM, hSalS);
        }
    }
}

```

```

let diff = salida - entrada;
if (diff < 0) return "Error";

const horas = Math.floor(diff / 3600000);
const minutos = Math.floor((diff % 3600000) / 60000);
const segundos = Math.floor((diff % 60000) / 1000);

return `${horas}h ${minutos}m ${segundos}s`;
}

// Construcción dinámica de la tabla
let tablaHTML =
    <table id="tablaBitacora" border="1" cellpadding="5"
cellspacing="0">
    <thead>
        <tr>
            <th>Usuario</th>
            <th>Hora Entrada</th>
            <th>Hora Salida</th>
            <th>Fecha</th>
            <th>Duración</th>
        </tr>
    </thead>
    <tbody>
        ${json.data
            .map((b) => {
                const duracion = calcularDuracion(b.horaEntrada,
b.horaSalida);
                return `
                    <tr>
                        <td>${b.nombreUsuario}</td>
                        <td>${b.horaEntrada}</td>
                        <td>${b.horaSalida}</td>
                        <td>${b.fecha}</td>
                        <td>${duracion}</td>
                    </tr>
                `;
            })
            .join("")
        }
    </tbody>
</table>
`;

contenedor.innerHTML = tablaHTML;
} catch (e) {
    console.error("X Error al cargar bitácora:", e);
    const contenedor = document.getElementById("contenedorTabla");
}

```

```

        if (contenedor) contenedor.innerHTML = "<p>Error al cargar los
datos.</p>";
    }
}

/***
 * Exporta la tabla de bitácora a PDF usando jsPDF y autoTable.
*/
function exportToPDF() {
    const { jsPDF } = window.jsPDF;
    const doc = new jsPDF();
    doc.text("Bitácora de Usuarios", 10, 10);

    const table = document.getElementById("tablaBitacora");
    if (!table) return;

    const rows = [];
    for (let i = 1; i < table.rows.length; i++) {
        const row = [];
        for (let j = 0; j < table.rows[i].cells.length; j++) {
            row.push(table.rows[i].cells[j].innerText);
        }
        rows.push(row);
    }

    doc.autoTable({
        head: [...Array.from(table.rows[0].cells).map((th) =>
th.innerText)],
        body: rows,
        startY: 20,
    });

    doc.save("bitacora.pdf");
}

/***
 * Exporta la tabla de bitácora a Excel (formato .xls).
*/
function exportToExcel() {
    const table = document.getElementById("tablaBitacora");
    if (!table) return;

    const html = table.outerHTML.replace(/ /g, "%20");
    const a = document.createElement("a");
    a.href = "data:application/vnd.ms-excel," + html;
    a.download = "bitacora.xls";
    a.click();
}

```



```

// Formato exacto AAAA-MM-DD
if (!/^\d{4}-\d{2}-\d{2}$/.test(fecha)) return false;

const [y, m, d] = fecha.split("-").map(Number);

// Fecha real (evita 2024-02-31, etc.)
const dtUTC = new Date(Date.UTC(y, m - 1, d));
const esReal =
    dtUTC.getUTCFullYear() === y &&
    dtUTC.getUTCMonth() + 1 === m &&
    dtUTC.getUTCDate() === d;
if (!esReal) return false;

const hoy = new Date();
const min = new Date("1900-01-01T00:00:00Z");
const dtLocal = new Date(y, m - 1, d); // para comparar con hoy en
local

// Rango permitido: [1900-01-01, hoy]
if (dtUTC < min || dtLocal > hoy) return false;

// Edad 0-120
const edad =
    hoy.getFullYear() - y -
    ((hoy.getMonth() + 1) < m || ((hoy.getMonth() + 1) === m &&
hoy.getDate() < d)) ? 1 : 0);

return edad >= 0 && edad <= 120;
},
esTextoLibreValido: (texto) => {
    const regex = /^[a-zA-ZÁÉÍÓÚáéíóúñÑ0-9\s.,\-\-]{0,100}$/;
    return regex.test(texto);
},
};

/***
 * Carga todos los clientes desde el servidor y los muestra en la tabla.
 */
const cargarClientes = async () => {
try {
    lista.innerHTML = `<div class="text-center p-3">
        <div class="spinner-border text-primary" role="status"></div>
        <p class="mt-2">Cargando clientes...</p>
    </div>`;
}

```

```

    const res = await
fetch("/CRM_INT/CRM/controller/ClienteController.php", {
    method: "POST",
    headers: { "Content-Type": "application/x-www-form-urlencoded" },
    body: "action=readAll",
});
const response = await res.json();
if (response.success && response.data) {
    mostrarClientes(response.data);
} else {
    lista.innerHTML =
        '<div class="alert alert-warning">No se pudieron cargar los
        clientes</div>';
}
} catch {
    lista.innerHTML = '<div class="alert alert-danger">Error de
    conexión</div>';
}
};

const mostrarClientes = (clientes) => {
    clientesActuales = clientes; // Guardar para eventos de la tabla

    // Si es la primera vez o estamos cargando todos los clientes,
    actualizar originales
    if (clientes.length > 0 && clientesOriginales.length === 0) {
        clientesOriginales = [...clientes];
    }

    if (clientes.length === 0) {
        lista.innerHTML =
            '<div class="alert alert-info">No hay clientes registrados</div>';
        return;
    }

    lista.innerHTML = `
        <table class="table table-striped table-hover mt-4"
        id="tablaClientes">
            <thead class="table-dark">
                <tr>
                    <th>ID</th><th>Cédula</th><th>Nombre</th><th>Correo</th><
                    th>Teléfono</th><th>Lugar</th><th>Cumpleaños</th><th>Acciones</th>
                </tr>
            </thead>
            <tbody>
                ${clientes
                    .map(
                        (cliente) => `
```

```

        <tr class="fila-cliente" data-id="${cliente.id}">
            <td>${cliente.id}</td>
            <td>${cliente.cedula}</td>
            <td>${cliente.nombre.toUpperCase()}</td>
            <td class="celda-correo">${cliente.correo ? cliente.correo.toUpperCase() : ""}</td>

            <td>${cliente.telefono}</td>
            <td>${cliente.lugarResidencia
                ? cliente.lugarResidencia.toUpperCase()
                : ""}</td>
            <td>${cliente.fechaCumpleanos}</td>
            <td>

<div class="acciones-cliente d-grid gap-1" style="grid-template-columns: repeat(2, 1fr); display: grid;">
    <button class="btn btn-sm btn-warning" onclick="editarCliente('${cliente.id}')">Editar</button>
    <button class="btn btn-sm btn-danger" onclick="eliminarCliente('${cliente.id}')">Eliminar</button>
    <button class="btn btn-sm" style="background-color: #111; color: #FFD600; border: none;" onclick="abrirModalReasignar(${cliente.id}, `#${cliente.nombre}\`, `${cliente.cedula}`)">Reasignar</button>
    <button class="btn btn-sm btn-info text-white" onclick="mostrarInfoCliente(${cliente.id})">Info</button>
</div>
</td>

        </tr>
        `

    )
    .join("")}
</tbody>
</table>
`;

// ---- EVENTO CLICK EN FILA (autollenar formulario) ----
document.querySelectorAll(".fila-cliente").forEach((fila) => {
    fila.addEventListener("click", function (e) {
        // Evita que dispare si se da click en un botón dentro de la fila

```

```

        if (e.target.tagName === "BUTTON" || e.target.closest("button"))
return;

        const id = this.getAttribute("data-id");
        const cliente = clientesActuales.find((c) => c.id == id);
        if (cliente) {
            // Autollenar formulario (mismos campos que editarCliente, pero
instantáneo)
            campoId.value = cliente.id;
            form.cedula.value = cliente.cedula;
            form.nombre.value = cliente.nombre;
            form.correo.value = cliente.correo || "";
            form.telefono.value = cliente.telefono;
            form.lugarResidencia.value = cliente.lugarResidencia || "";
            form.fechaCumpleanos.value = cliente.fechaCumpleanos;
            submitBtn.textContent = "Actualizar Cliente";
            submitBtn.className = "btn btn-warning";
            editandoId = cliente.id;
            campoId.disabled = true;
            cancelBtn.style.display = "inline-block";
            form.cedula.focus();
        }
    });
});
};

// Función para inicializar o actualizar la lista completa de clientes
const cargarClientesCompletos = (clientes) => {
    clientesOriginales = [...clientes]; // Guardar copia completa
    mostrarClientes(clientes);
};

// Buscador en tiempo real por número de tarjeta (ID) o nombre
document.addEventListener("DOMContentLoaded", () => {
    const buscador = document.getElementById("buscadorClientes");
    if (!buscador) return;

    buscador.placeholder = "Buscar por Tarjeta (ID) o Cédula";

    buscador.addEventListener("input", function () {
        const valor = this.value.trim();
        if (!valor) { mostrarClientes(clientesOriginales); return; }

        const digitos = valor.replace(/\D/g, ""); // usar solo números

        const filtrados = clientesOriginales.filter((c) => {
            const idMatch = c.id && String(c.id).includes(digitos);
            const cedulaMatch =

```

```

        c.cedula &&
        c.cedula.toString().replace(/\D/g, "").includes(digitos);
        return idMatch || cedulaMatch;
    });

    mostrarClientes(filtrados);
});

});

/***
 * Carga los datos del cliente seleccionado para edición.
 */
const editarCliente = async (id) => {
    try {
        const res = await
fetch("/CRM_INT/CRM/controller/ClienteController.php", {
            method: "POST",
            headers: { "Content-Type": "application/x-www-form-urlencoded" },
            body: `action=read&id=${id}`,
        });
        const response = await res.json();
        if (response.success && response.data) {
            const cliente = response.data;
            campoId.value = cliente.id;
            form.cedula.value = cliente.cedula;
            form.nombre.value = cliente.nombre;
            form.correo.value = cliente.correo;
            form.telefono.value = cliente.telefono;
            form.lugarResidencia.value = cliente.lugarResidencia;
            form.fechaCumpleanos.value = cliente.fechaCumpleanos;
            submitBtn.textContent = "Actualizar Cliente";
            submitBtn.className = "btn btn-warning";
            editandoId = cliente.id;
            campoId.disabled = true;
            cancelBtn.style.display = "inline-block";

            // Llevar el focus a cédula
            form.cedula.focus();
        }
    } catch {
        alert("Error al cargar los datos del cliente");
    }
};

/***
 * Elimina un cliente mediante confirmación.
 */
const eliminarCliente = async (id) => {

```

```

    if (!confirm("¿Estás seguro de que quieres eliminar este cliente?"))
return;
try {
    const res = await
fetch("/CRM_INT/CRM/controller/ClienteController.php", {
    method: "POST",
    headers: { "Content-Type": "application/x-www-form-urlencoded" },
    body: `action=delete&id=${id}`,
});
const response = await res.json();
alert(response.message);
if (response.success) {
    cargarClientes();

    // 🔔 ACTUALIZAR BADGE DE CUMPLEAÑOS SI EXISTE LA FUNCIÓN
    if (window.actualizarCumpleBadgeSidebar) {
        window.actualizarCumpleBadgeSidebar();
    }
}
} catch {
    alert("Error al eliminar el cliente");
}
};

/***
 * Restaura el formulario a su estado inicial.
 */
const cancelarEdicion = () => {
editandoId = null;
form.reset();
campoId.disabled = true;
campoId.value = "";
submitBtn.textContent = "Guardar Cliente";
submitBtn.className = "btn btn-primary";
cancelBtn.style.display = "none";

// Limpiar mensajes de validación personalizados
const inputs = form.querySelectorAll("input");
inputs.forEach((input) => input.setCustomValidity(""));
};

// Manejador del envío del formulario
form.onsubmit = async (e) => {
e.preventDefault();

// Obtener valores del formulario
const cedula = form.cedula.value.trim();
const nombre = form.nombre.value.trim();

```

```

const correo = form.correo.value.trim();
const telefono = form.telefono.value.trim();
const telefonoLimpio = telefono.replace("-", "");
const lugarResidencia = form.lugarResidencia.value.trim();
const fechaCumpleanos = form.fechaCumpleanos.value;

// Validaciones
if (!cedula || !nombre || !telefono || !lugarResidencia ||
!fechaCumpleanos) {
    alert("Todos los campos son obligatorios (excepto el correo).");
    return;
}

if (!/^\\d{9}$|\\d{14}\\.test(cedula)) {
    alert("La cédula debe tener exactamente 9 o 14 dígitos numéricos.");
    form.cedula.focus();
    return;
}

if (telefonoLimpio.length !== 8) {
    alert("El teléfono debe tener exactamente 8 dígitos.");
    form.telefono.focus();
    return;
}

if (!validaciones.esNombreValido(nombre)) {
    alert("El nombre solo puede contener letras y espacios.");
    form.nombre.focus();
    return;
}

if (nombre.length > 45) {
    alert("El nombre no puede tener más de 45 caracteres.");
    form.nombre.focus();
    return;
}

if (correo && !validaciones.esEmailValido(correo)) {
    alert("Por favor ingrese un correo electrónico válido.");
    form.correo.focus();
    return;
}

if (!validaciones.esFechaValida(fechaCumpleanos)) {
    alert("Por favor ingrese una fecha de cumpleaños válida.");
    form.fechaCumpleanos.focus();
    return;
}

```

```

if (form.alergias && form.alergias.value.trim()) {
    const alergias = form.alergias.value.trim();
    if (!validaciones.esTextoLibreValido(alergias)) {
        alert(
            "Las alergias deben contener solo letras, números, espacios y
puntuación básica (máx. 100 caracteres)."
        );
        form.alergias.focus();
        return;
    }
}

if (form.gustosEspeciales && form.gustosEspeciales.value.trim()) {
    const gustos = form.gustosEspeciales.value.trim();
    if (!validaciones.esTextoLibreValido(gustos)) {
        alert(
            "Los gustos especiales deben contener solo letras, números,
espacios y puntuación básica (máx. 100 caracteres)."
        );
        form.gustosEspeciales.focus();
        return;
    }
}

const formData = new FormData(form);
formData.set("telefono", telefonoLimpio);
const action = editandoId ? "update" : "create";
formData.append("action", action);
if (editandoId) formData.append("id", editandoId);

try {
    const res = await
fetch("/CRM_INT/CRM/controller/ClienteController.php", {
    method: "POST",
    body: formData,
});
    const response = await res.json();

    if (!response.success) {
        alert(response.message || "Error al guardar el cliente.");
        return;
    }

    let msg = response.message || "Cliente guardado correctamente";
    if (response.debug) {
        msg += "\n\nDEBUG:\n" + response.debug;
    }
}

```

```

        alert(msg);
        cancelarEdicion();
        cargarClientes();

        // ♦ ACTUALIZAR BADGE DE CUMPLEAÑOS SI EXISTE LA FUNCIÓN
        if (window.actualizarCumpleBadgeSidebar) {
            window.actualizarCumpleBadgeSidebar();
        }
    } catch (error) {
        console.error("Error real al guardar cliente:", error);
        alert("Error de conexión al guardar");
    }
};

// Inicialización cuando se carga la vista
document.addEventListener("DOMContentLoaded", () => {
    campoId.disabled = true;
    cancelBtn.style.display = "none";

    // Configurar validaciones y conversiones en tiempo real
    const cedulaInput = form.cedula;
    const nombreInput = form.nombre;
    const correoInput = form.correo;
    const telefonoInput = form.telefono;
    const lugarInput = form.lugarResidencia;
    const fechaInput = form.fechaCumpleanos;

    // Validaciones en tiempo real para cédula
    if (cedulaInput) {
        cedulaInput.addEventListener("input", function () {
            this.value = this.value.replace(/\D/g, ""); // Solo números

            // Limitar a 14 dígitos (máximo permitido)
            if (this.value.length > 14) {
                this.value = this.value.slice(0, 14);
            }

            this.setCustomValidity("");
        });

        cedulaInput.addEventListener("invalid", function () {
            this.setCustomValidity(
                "Por favor ingrese una cédula válida (9 o 14 dígitos)"
            );
        });
    }
}

```

```

// Conversión automática a mayúsculas para nombre y límite de
caracteres
if (nombreInput) {
    nombreInput.addEventListener("input", function () {
        this.value = this.value.toUpperCase();

        // Limitar a 50 caracteres
        if (this.value.length > 50) {
            this.value = this.value.slice(0, 50);
        }

        this.setCustomValidity("");
    });

    nombreInput.addEventListener("invalid", function () {
        this.setCustomValidity("Por favor ingrese un nombre válido");
    });
}

// Conversión automática a mayúsculas para correo
if (correoInput) {
    correoInput.addEventListener("input", function () {
        this.value = this.value.toUpperCase();
        this.setCustomValidity("");
    });

    correoInput.addEventListener("invalid", function () {
        this.setCustomValidity("Por favor ingrese un correo electrónico
válido");
    });
}

// Validaciones para teléfono
// Validaciones para teléfono con guion automático
if (telefonoInput) {
    telefonoInput.addEventListener("input", function () {
        let valor = this.value.replace(/\D/g, ""); // Solo números

        // Limitar a 8 dígitos
        if (valor.length > 8) {
            valor = valor.slice(0, 8);
        }

        // Agregar guion en la posición 4 si hay más de 4 dígitos
        if (valor.length > 4) {
            this.value = valor.slice(0, 4) + "-" + valor.slice(4);
        } else {
            this.value = valor;
        }
    });
}

```

```

    }

    this.setCustomValidity("");
});

telefonoInput.addEventListener("invalid", function () {
    this.setCustomValidity(
        "Por favor ingrese un teléfono válido de 8 dígitos"
    );
});

// Conversión automática a mayúsculas y validación para lugar de
residencia
if (lugarInput) {
    lugarInput.addEventListener("input", function () {
        this.value = this.value.toUpperCase();

        // Limitar a 50 caracteres
        if (this.value.length > 50) {
            this.value = this.value.slice(0, 50);
        }

        this.setCustomValidity("");
    });

    lugarInput.addEventListener("invalid", function () {
        this.setCustomValidity("Por favor ingrese el lugar de residencia");
    });
}

// Validación para fecha
if (fechaInput) {
    fechaInput.addEventListener("invalid", function () {
        this.setCustomValidity(
            "Por favor seleccione una fecha de cumpleaños válida"
        );
    });

    fechaInput.addEventListener("input", function () {
        this.setCustomValidity("");
    });
}

/***
 * Funciones para reasignación de códigos
 */

```

```

// Variables globales para el modal de reasignación
let clienteParaReasignar = null;

/**
 * Abre el modal de reasignación de código
 */
const abrirModalReasignar = (idCliente, nombreCliente, cedulaCliente) =>
{
    console.log("Abriendo modal para cliente:", idCliente);

    clienteParaReasignar = {
        id: idCliente,
        nombre: nombreCliente,
        cedula: cedulaCliente,
    };

    // Actualizar información del cliente en el modal
    const modalClienteNombre =
document.getElementById("modalClienteNombre");
    const modalClienteCedula =
document.getElementById("modalClienteCedula");
    const modalClienteId = document.getElementById("modalClienteId");
    const motivoSelect = document.getElementById("motivoSelect");
    const motivoReasignacion =
document.getElementById("motivoReasignacion");

    if (modalClienteNombre) modalClienteNombre.textContent = nombreCliente;
    if (modalClienteCedula) modalClienteCedula.textContent = cedulaCliente;
    if (modalClienteId) modalClienteId.textContent = idCliente;

    // Limpiar y resetear los campos de motivo
    if (motivoSelect) motivoSelect.selectedIndex = 0;
    if (motivoReasignacion) {
        motivoReasignacion.value = "";
        motivoReasignacion.style.display = "none";
    }

    // Verificar si el modal existe
    const modalElement = document.getElementById("modalReasignar");
    if (!modalElement) {
        console.error("Modal no encontrado");
        alert("Error: No se pudo encontrar el modal");
        return;
    }

    // Mostrar el modal
    try {

```

```

const modal = new bootstrap.Modal(modalElement, {
    backdrop: false, // Sin fondo oscuro
    keyboard: true, // Permite cerrar con ESC
    focus: true, // Enfoca el modal al abrirse
});
modal.show();
} catch (error) {
    console.error("Error al abrir modal:", error);
    // Fallback: usar jQuery si está disponible
    if (typeof $ !== "undefined") {
        $("#modalReasignar").modal("show");
    } else {
        alert(
            "Error al abrir el modal. Por favor recarga la página e intenta de nuevo."
        );
    }
};

/***
 * Maneja la selección del motivo de reasignación
 */
const manejarSeleccionMotivo = () => {
    const motivoSelect = document.getElementById("motivoSelect");
    const motivoTextarea = document.getElementById("motivoReasignacion");

    const valorSeleccionado = motivoSelect.value;

    if (valorSeleccionado === "otro") {
        // Mostrar textarea para motivo personalizado
        motivoTextarea.style.display = "block";
        motivoTextarea.value = "";
        motivoTextarea.placeholder = "Especifique el motivo...";
        motivoTextarea.focus();
    } else if (valorSeleccionado !== "") {
        // Ocultar textarea y usar el valor seleccionado
        motivoTextarea.style.display = "none";
        motivoTextarea.value = valorSeleccionado;
    } else {
        // No hay selección, ocultar textarea
        motivoTextarea.style.display = "none";
        motivoTextarea.value = "";
    }
};

/***
 * Procesa la reasignación del código
*/

```

```

*/
const procesarReasignacion = async () => {
  try {
    if (!clienteParaReasignar || !clienteParaReasignar.id) {
      alert("Error: No se ha seleccionado un cliente válido");
      return;
    }

    // Obtener el motivo correcto
    const motivoSelect = document.getElementById("motivoSelect");
    let motivo = "";

    if (motivoSelect && motivoSelect.value) {
      if (motivoSelect.value === "otro") {
        const motivoReasignacion =
          document.getElementById("motivoReasignacion");
        motivo = motivoReasignacion ? motivoReasignacion.value.trim() :
        "";
      }

      if (!motivo) {
        alert("Por favor especifica el motivo de reasignación");
        return;
      }
    } else {
      motivo = motivoSelect.value;
    }
  } else {
    alert("Por favor selecciona un motivo para la reasignación");
    return;
  }

  const btnConfirmar =
document.getElementById("btnConfirmarReasignacion");
  const textoOriginal = btnConfirmar.textContent;
  btnConfirmar.disabled = true;
  btnConfirmar.textContent = "Procesando...";

  // El procedimiento ahora se encarga de encontrar el código activo
  internamente
  const res = await
fetch("/CRM_INT/CRM/controller/ClienteController.php", {
  method: "POST",
  headers: { "Content-Type": "application/x-www-form-urlencoded" },
  body: `action=reasssignCode&idCliente=${
    clienteParaReasignar.id
    }&motivo=${encodeURIComponent(motivo)}`,
});

```

```

const response = await res.json();

btnConfirmar.disabled = false;
btnConfirmar.textContent = textoOriginal;

if (response.success) {
    alert(`Código reasignado exitosamente.\nNuevo código:
${response.nuevoCodigo}`);
}

// Cerrar el modal
const modal = bootstrap.Modal.getInstance(
    document.getElementById("modalReasignar")
);
modal.hide();

// Recargar la lista de clientes Y el historial de reasignaciones
cargarClientes();
cargarHistorialReasignaciones(true); // Manual después de
reasignación
} else {
    alert(`Error al reasignar código: ${response.message}`);
}
} catch (error) {
    console.error("Error:", error);
    alert("Error de conexión al procesar la reasignación");

    const btnConfirmar =
document.getElementById("btnConfirmarReasignacion");
    btnConfirmar.disabled = false;
    btnConfirmar.textContent = "Confirmar";
}
};

// Inicialización cuando se carga la vista
document.addEventListener("DOMContentLoaded", () => {
    cargarClientes();

    // Verificar que el modal existe al cargar la página
    const modal = document.getElementById("modalReasignar");
    if (modal) {
        console.log("Modal de reasignación encontrado correctamente");

        // Agregar evento para cerrar modal al hacer clic en el fondo
        modal.addEventListener("click", function (event) {
            if (event.target === modal) {
                const bootstrapModal = bootstrap.Modal.getInstance(modal);

```

```

        if (bootstrapModal) {
            bootstrapModal.hide();
        }
    });

    // Agregar evento para cerrar con ESC
    document.addEventListener("keydown", function (event) {
        if (event.key === "Escape") {
            const bootstrapModal = bootstrap.Modal.getInstance(modal);
            if (bootstrapModal && modal.classList.contains("show")) {
                bootstrapModal.hide();
            }
        }
    });
} else {
    console.error("Modal de reasignación NO encontrado en el DOM");
}
});

/***
 * Actualiza el estado visual de la actualización automática
 */
const actualizarEstadoVisualizacion = (estado) => {
    const estadoElement = document.getElementById("estadoActualizacion");
    if (estadoElement) {
        switch (estado) {
            case "cargando":
                estadoElement.innerHTML =
                    '<i class="fas fa-spinner fa-spin me-1"></i>Actualizando...';
                break;
            case "activo":
                estadoElement.innerHTML = "Actualización automática activa";
                break;
            case "error":
                estadoElement.innerHTML =
                    '<i class="fas fa-exclamation-triangle me-1"></i>Error en
actualización';
                break;
            default:
                estadoElement.innerHTML = "Actualización automática activa";
        }
    }
};

/***
 * Carga el historial de reasignaciones desde el servidor
*/

```

```

const cargarHistorialReasignaciones = async (esManual = false) => {
  const contenedor = document.getElementById("historialReasignaciones");

  // Solo mostrar spinner en carga manual, no en automática
  const esActualizacionManual = esManual || !intervaloActualizacion;

  if (esActualizacionManual) {
    contenedor.innerHTML = `<div class="text-center p-3">
      <div class="spinner-border text-primary" role="status"></div>
      <p class="mt-2">Cargando historial...</p>
    </div>`;
  } else {
    // Para actualización automática, solo cambiar el estado
    actualizarEstadoVisualizacion("cargando");
  }

  try {
    // Agregar timestamp para evitar cache del navegador
    const timestamp = new Date().getTime();
    const res = await
    fetch("/CRM_INT/CRM/controller/ClienteController.php", {
      method: "POST",
      headers: {
        "Content-Type": "application/x-www-form-urlencoded",
        "Cache-Control": "no-cache",
      },
      body: `action=getHistorialReasignaciones&timestamp=${timestamp}`,
    });

    const response = await res.json();

    if (response.success && response.data) {
      // Procesar los datos para obtener el historial de reasignaciones
      const historialProcesado = procesarHistorialReasignaciones(
        response.data.codigos,
        response.data.clientes
      );
      mostrarHistorialReasignaciones(historialProcesado);

      // Actualizar estado visual
      actualizarEstadoVisualizacion("activo");
    } else {
      contenedor.innerHTML =
        '<div class="alert alert-warning">No se pudieron cargar los
      registros de reasignaciones</div>';
      actualizarEstadoVisualizacion("error");
    }
  } catch (error) {

```

```

        console.error("Error cargando historial:", error);
        contenedor.innerHTML =
            '<div class="alert alert-danger">Error de conexión al cargar el
historial</div>';
            actualizarEstadoVisualizacion("error");
        }
    };

    /**
     * Procesa los datos de códigos para extraer el historial de
reasignaciones
    */
const procesarHistorialReasignaciones = (codigos, clientes) => {
    const ahora = new Date().toLocaleTimeString();
    console.log(`[${ahora}] Procesando historial de reasignaciones...`);
    console.log("Total códigos recibidos:", codigos.length);
    console.log("Total clientes recibidos:", clientes.length);

    // Crear un mapa de clientes para búsqueda rápida
    const clientesMap = {};
    clientes.forEach((cliente) => {
        clientesMap[cliente.id] = cliente;
    });

    // Agrupar por cliente y separar códigos activos/inactivos
    const agrupadoPorCliente = {};
    codigos.forEach((codigo) => {
        const idCliente = codigo.idCliente;
        const cliente = clientesMap[idCliente];
        if (!agrupadoPorCliente[idCliente]) {
            agrupadoPorCliente[idCliente] = {
                cliente: {
                    id: idCliente,
                    nombre: cliente ? cliente.nombre : "Cliente no encontrado",
                    cedula: cliente ? cliente.cedula : "-",
                },
                codigoActivo: null,
                reasignaciones: [],
                totalReasignaciones: 0,
            };
        }
        if (codigo.estado === "Activo") {
            agrupadoPorCliente[idCliente].codigoActivo = {
                codigoBarra: codigo.codigoBarra,
                cantImpresiones: codigo.cantImpresiones || 0,
                fechaRegistro: codigo.fechaRegistro,
            };
        } else if (

```

```

        codigo.estado === "Inactivo" &&
        codigo.motivoCambio &&
        codigo.motivoCambio.trim() !== ""
    ) {
        agrupadoPorCliente[idCliente].reasignaciones.push({
            codigoBarra: codigo.codigoBarra,
            cantImpresiones: codigo.cantImpresiones || 0,
            motivoCambio: codigo.motivoCambio,
            fechaCambio: codigo.fechaCambio,
            fechaRegistro: codigo.fechaRegistro,
        });
        agrupadoPorCliente[idCliente].totalReasignaciones++;
    }
});
// Ordenar las reasignaciones por fecha más reciente primero
Object.values(agrupadoPorCliente).forEach((cliente) => {
    cliente.reasignaciones.sort((a, b) => {
        const fechaA = new Date(a.fechaCambio || a.fechaRegistro || "1970-01-01");
        const fechaB = new Date(b.fechaCambio || b.fechaRegistro || "1970-01-01");
        if (fechaA.getTime() === fechaB.getTime()) {
            return (b.cantImpresiones || 0) - (a.cantImpresiones || 0);
        }
        return fechaB - fechaA;
    });
    // Para ordenar la lista principal por la fecha más reciente (activo o inactivo)
    let fechaMasReciente = null;
    if (cliente.codigoActivo) {
        fechaMasReciente = new Date(
            cliente.codigoActivo.fechaRegistro || "1970-01-01"
        );
    } else if (cliente.reasignaciones.length > 0) {
        fechaMasReciente = new Date(
            cliente.reasignaciones[0].fechaCambio ||
            cliente.reasignaciones[0].fechaRegistro ||
            "1970-01-01"
        );
    } else {
        fechaMasReciente = new Date("1970-01-01");
    }
    cliente.fechaMasReciente = fechaMasReciente;
});
// Ordenar clientes por fecha más reciente
const historialArray = Object.values(agrupadoPorCliente).sort(
    (a, b) => b.fechaMasReciente - a.fechaMasReciente
);

```

```

const ahora2 = new Date().toLocaleTimeString();
console.log(
  `[${ahora2}] Historial procesado:`,
  historialArray.length,
  "clientes con reasignaciones"
);
historialArray.forEach((cliente) => {
  console.log(
    `Cliente ${cliente.cliente.nombre}: ${cliente.totalReasignaciones}
reasignaciones`
  );
});
return historialArray;
};

/**
 * Muestra el historial de reasignaciones en una tabla
 */
const mostrarHistorialReasignaciones = (historialPorCliente) => {
  const contenedor = document.getElementById("historialReasignaciones");

  if (!historialPorCliente || historialPorCliente.length === 0) {
    contenedor.innerHTML =
      '<div class="alert alert-info">No hay reasignaciones
registradas</div>';
    return;
  }

  const formatearFecha = (fecha) => {
    if (!fecha) return "-";
    const date = new Date(fecha);
    return date.toLocaleDateString("es-CR", {
      year: "numeric",
      month: "2-digit",
      day: "2-digit",
    });
  };

  // Mostrar solo la fila del código activo por cliente
  const filasClientes = historialPorCliente
    .map((clienteData) => {
      const cliente = clienteData.cliente;
      const codigoActivo = clienteData.codigoActivo;
      if (!codigoActivo) return "";
      // Buscar la última reasignación (inactiva más reciente)
      let ultimoMotivo = "-";
      let ultimaFecha = "-";

```

```

        if (clienteData.reasignaciones && clienteData.reasignaciones.length
> 0) {
            const ultimaReasignacion = clienteData.reasignaciones[0];
            ultimoMotivo = ultimaReasignacion.motivoCambio || "-";
            ultimaFecha = formatearFecha(
                ultimaReasignacion.fechaCambio ||
                ultimaReasignacion.fechaRegistro
            );
        }
        return `
<tr class="fila-cliente">
    <td class="text-center">
        <strong>${cliente.id}</strong>
    </td>
    <td>
        <strong>${cliente.cedula || "-"}</strong>
    </td>
    <td>
        <strong>${cliente.nombre || "-"}</strong>
    </td>
    <td class="text-center">
        <code class="codigo-barra">${codigoActivo.codigoBarra}</code>
        <br>
        <small class="text-success">(activo)</small>
    </td>
    <td class="text-center">
        <span class="badge bg-warning text-dark">${
            clienteData.totalReasignaciones
        }</span>
    </td>
    <td class="text-center">
        <span class="badge bg-success">Activo</span>
    </td>
    <td class="motivo-cell">${ultimoMotivo}</td>
    <td class="text-center">
        <small class="text-muted">${ultimaFecha}</small>
    </td>
</tr>
`;
})
.join("");
}

contenedor.innerHTML = `
<table class="table table-striped table-hover mt-2">
    <thead class="table-dark">
        <tr>
            <th class="text-center">Tarjeta</th>
            <th>Cédula</th>

```

```

        <th>Cliente</th>
        <th class="text-center">Código</th>
        <th class="text-center">Total Reasignaciones</th>
        <th class="text-center">Estado</th>
        <th>Último Motivo</th>
        <th class="text-center">Fecha</th>
    </tr>
</thead>
<tbody>
    ${filasClientes}
</tbody>
</table>
`;
};

/**
 * Función para mostrar/ocultar detalles de reasignaciones de un cliente
 */
window.toggleDetalleReasignaciones = function (clienteIndex) {
    const detalles = document.querySelectorAll(`#detalle-${clienteIndex}`);
    const isVisible = detalles[0].style.display !== "none";

    detalles.forEach((detalle) => {
        detalle.style.display = isVisible ? "none" : "";
        if (!isVisible) {
            detalle.style.backgroundColor = "#f8f9fa";
        }
    });
};

/**
 * Actualización automática del historial cada 30 segundos
 */
let intervaloActualizacion = null;

const iniciarActualizacionAutomatica = () => {
    // Limpiar cualquier intervalo existente
    if (intervaloActualizacion) {
        clearInterval(intervaloActualizacion);
    }

    // Actualizar cada 30 segundos
    intervaloActualizacion = setInterval(() => {
        const ahora = new Date().toLocaleTimeString();
        console.log(`[${ahora}] Actualizando historial automáticamente...`);
        cargarHistorialReasignaciones();
    }, 30000); // 30 segundos
}

```

```

    console.log("Auto-refresh iniciado: cada 30 segundos");
};

const detenerActualizacionAutomatica = () => {
    if (intervaloActualizacion) {
        clearInterval(intervaloActualizacion);
        intervaloActualizacion = null;
    }
};
window.mostrarInfoCliente = (id) => {
    const cliente = clientesActuales.find((c) => c.id == id);
    if (!cliente) {
        Swal.fire({
            icon: "error",
            title: "Cliente no encontrado",
            text: "No se pudo encontrar el cliente seleccionado.",
        });
        return;
    }

    const alergias = cliente.alergias?.trim() || "Sin datos registrados";
    const gustos = cliente.gustosEspeciales?.trim() || "Sin datos registrados";

    Swal.fire({
        title: `👤 Información de ${cliente.nombre}`,
        html: `
            <div class="text-start">
                <strong>Alergias:</strong><br>
                <p>${alergias}</p>
                <strong>Gustos Especiales:</strong><br>
                <p>${gustos}</p>
            </div>
        `,
        icon: "info",
        confirmButtonText: "Cerrar",
        customClass: {
            popup: "text-left",
        },
    });
};

// Cargar historial al inicializar la página
document.addEventListener("DOMContentLoaded", function () {
    cargarHistorialReasignaciones(true); // Manual al cargar la página
    iniciarActualizacionAutomatica();

    // Detener actualización automática cuando se cambia de página
}

```

```

        window.addEventListener("beforeunload",
detenerActualizacionAutomatica);
});

document.addEventListener("DOMContentLoaded", () => {
    const cedulaInput = document.getElementById("clienteCedula");

    cedulaInput.addEventListener("input", function () {
        this.value = this.value.replace(/\D/g, "");
        if (this.value.length > 14) {
            this.value = this.value.slice(0, 14);
        }
    });
}

const alergiasInput = form.alergias;
const gustosInput = form.gustosEspeciales;

[alergiasInput, gustosInput].forEach((input) => {
    if (input) {
        input.addEventListener("input", function () {
            // Solo letras, números, espacios, comas, puntos, guiones
            this.value = this.value.replace(/[^a-zA-ZÁÉÍÓÚáéíóúññ0-9\s.,\-\]/g, "");

            // Limitar a 100 caracteres
            if (this.value.length > 100) {
                this.value = this.value.slice(0, 100);
            }

            this.setCustomValidity("");
        });
    }
});

const nombreInput = document.getElementById("clienteNombre");

nombreInput.addEventListener("input", function () {
    // Solo letras (mayúsculas, minúsculas) y espacios
    this.value = this.value.replace(/[^a-zA-ZÁÉÍÓÚáéíóúññ\s]/g, "");

    // Limita a 45 caracteres
    if (this.value.length > 45) {
        this.value = this.value.slice(0, 45);
    }
});

const cantones = [
    "LIBERIA, GUANACASTE",
    "NICOYA, GUANACASTE",
    "SANTA CRUZ, GUANACASTE",
]

```

"BAGACES, GUANACASTE",  
"CARRILLO, GUANACASTE",  
"CAÑAS, GUANACASTE",  
"ABANGARES, GUANACASTE",  
"TILARÁN, GUANACASTE",  
"NANDAYURE, GUANACASTE",  
"LA CRUZ, GUANACASTE",  
"HOJANCHA, GUANACASTE",  
"PUNTARENAS, PUNTARENAS",  
"ESPARZA, PUNTARENAS",  
"BUENOS AIRES, PUNTARENAS",  
"MONTES DE ORO, PUNTARENAS",  
"OSA, PUNTARENAS",  
"AGUIRRE, PUNTARENAS",  
"GOLFITO, PUNTARENAS",  
"COTO BRUS, PUNTARENAS",  
"PARAÍSO, PUNTARENAS",  
"CORREDORES, PUNTARENAS",  
"GARABITO, PUNTARENAS",  
"SAN JOSÉ, SAN JOSÉ",  
"ESCAZÚ, SAN JOSÉ",  
"DESAMPARADOS, SAN JOSÉ",  
"PURISCAL, SAN JOSÉ",  
"TARAZU, SAN JOSÉ",  
"ASERRÍ, SAN JOSÉ",  
"MORA, SAN JOSÉ",  
"GOICOECHEA, SAN JOSÉ",  
"SANTA ANA, SAN JOSÉ",  
"ALAJUELITA, SAN JOSÉ",  
"VÁSQUEZ DE CORONADO, SAN JOSÉ",  
"ACOSTA, SAN JOSÉ",  
"TIBÁS, SAN JOSÉ",  
"MORAVIA, SAN JOSÉ",  
"MONTES DE OCA, SAN JOSÉ",  
"TURRUBARES, SAN JOSÉ",  
"DOTA, SAN JOSÉ",  
"CURRIDABAT, SAN JOSÉ",  
"PÉREZ ZELEDÓN, SAN JOSÉ",  
"LEÓN CORTÉS, SAN JOSÉ",  
"ALAJUELA, ALAJUELA",  
"SAN RAMÓN, ALAJUELA",  
"GRECIA, ALAJUELA",  
"SAN MATEO, ALAJUELA",  
"ATENAS, ALAJUELA",  
"NARANJO, ALAJUELA",  
"PALMARES, ALAJUELA",  
"POÁS, ALAJUELA",  
"OROTINA, ALAJUELA",

```

    "SAN CARLOS, ALAJUELA",
    "ZARCERO, ALAJUELA",
    "VALVERDE VEGA, ALAJUELA",
    "UPALA, ALAJUELA",
    "LOS CHILES, ALAJUELA",
    "GUATUSO, ALAJUELA",
    "RIO CUARTO, ALAJUELA",
    "HEREDIA, HEREDIA",
    "BARVA, HEREDIA",
    "SANTO DOMINGO, HEREDIA",
    "SANTA BÁRBARA, HEREDIA",
    "SAN RAFAEL, HEREDIA",
    "SAN ISIDRO, HEREDIA",
    "BELÉN, HEREDIA",
    "FLORES, HEREDIA",
    "SARAPIQUÍ, HEREDIA",
    "CARTAGO, CARTAGO",
    "PARAÍSO, CARTAGO",
    "LA UNIÓN, CARTAGO",
    "JIMÉNEZ, CARTAGO",
    "TURRIALBA, CARTAGO",
    "ALVARADO, CARTAGO",
    "OREAMUNO, CARTAGO",
    "EL GUARCO, CARTAGO",
    "LIMÓN, LIMÓN",
    "POCOCÍ, LIMÓN",
    "SIQUIRRES, LIMÓN",
    "TALAMANCA, LIMÓN",
    "MATINA, LIMÓN",
    "GUÁCIMO, LIMÓN",
];
const datalist = document.getElementById("listaCantones");
cantones.forEach((canton) => {
  const option = document.createElement("option");
  option.value = canton;
  datalist.appendChild(option);
});
document.addEventListener("DOMContentLoaded", () => {
  const f = document.getElementById("clienteFecha"); // <- tu id en la
vista
  if (f) {
    const hoy = new Date().toISOString().slice(0, 10);
    f.setAttribute("min", "1900-01-01");
    f.setAttribute("max", hoy);
  }
});

```

```
});
```

## |   |— Código.js

```
let codigosGlobal = [] // Para filtro de búsqueda en vivo
let isScanning = false // Para detectar si se está usando el lector de
código de barras

/***
 * Carga todos los códigos desde el backend
 */
const cargarCódigos = async () => {
    const contenedor = document.getElementById("codigoLista");
    contenedor.innerHTML = `<div class="text-center p-3">
        <div class="spinner-border text-primary" role="status"></div>
        <p class="mt-2">Cargando códigos...</p>
    </div>`;
    try {
        const res = await
fetch("/CRM_INT/CRM/controller/CodigoController.php", {
            method: "POST",
            headers: { "Content-Type": "application/x-www-form-
urlencoded" },
            body: "action=readAll"
        });
        const response = await res.json();
        if (response.success && response.data) {
            console.log('Datos recibidos del backend:', response.data);
            console.log('Primer registro:', response.data[0]);

            // Filtrar solo códigos activos para el array global
            codigosGlobal = response.data.filter(código => código.estado
== 'Activo');
            console.log('Códigos activos filtrados:', codigosGlobal);

            mostrarCódigos(codigosGlobal);
        } else {
            contenedor.innerHTML = '<div class="alert alert-warning">No
se pudieron cargar los códigos</div>';
        }
    } catch {
        contenedor.innerHTML = '<div class="alert alert-danger">Error de
conexión</div>';
    }
};

/***
 * Genera un código de barras usando el número de tarjeta
*/
```

```

/*
function generarCodigoBarras(numeroTarjeta, elementoId) {
    try {
        const codigoParaBarras = String(numeroTarjeta).padStart(8, '0');
// Mínimo 8 dígitos
        JsBarcode(`#${elementoId}`, codigoParaBarras, {
            format: "CODE128",
            width: 2,
            height: 50,
            displayValue: true,
            fontSize: 12,
            margin: 5
        });
    } catch (error) {
        console.error('Error generando código de barras:', error);
        document.getElementById(elementoId).innerHTML = `<span
class="text-danger">Error: ${numeroTarjeta}</span>`;
    }
}

/**
 * Genera un código de barras para impresión (tamaño tarjeta) con mayor
calidad
*/
function generarCodigoBarrasParaImpresion(numeroTarjeta) {
    const canvas = document.createElement('canvas');
    const codigoParaBarras = String(numeroTarjeta).padStart(8, '0');
    JsBarcode(canvas, codigoParaBarras, {
        format: "CODE128",
        width: 2,
        height: 40,
        displayValue: true,
        fontSize: 12,
        margin: 3,
        background: "#ffffff",
        lineColor: "#000000",
        textAlign: "center",
        textPosition: "bottom",
        fontOptions: "bold"
    });
    const highResCanvas = document.createElement('canvas');
    const ctx = highResCanvas.getContext('2d');
    const scale = 2;
    highResCanvas.width = canvas.width * scale;
    highResCanvas.height = canvas.height * scale;
    ctx.imageSmoothingEnabled = false;
    ctx.webkitImageSmoothingEnabled = false;
    ctx.mozImageSmoothingEnabled = false;
}

```

```

        ctx.msImageSmoothingEnabled = false;
        ctx.drawImage(canvas, 0, 0, canvas.width * scale, canvas.height * scale);
        return highResCanvas;
    }

    /**
     * Imprime código de barras en PDF con alta calidad
     */
    async function imprimirPDF(numeroTarjeta, barcodeId) {
        try {
            const canvas = generarCodigoBarrasParaImpresion(numeroTarjeta);
            const imgData = canvas.toDataURL('image/png', 1.0);
            const { jsPDF } = window.jspdf;
            const doc = new jsPDF({
                orientation: 'landscape',
                unit: 'mm',
                format: [85.6, 53.98]
            });
            const imgWidth = 70;
            const imgHeight = 25;
            const x = (85.6 - imgWidth) / 2;
            const y = (53.98 - imgHeight) / 2;
            doc.addImage(imgData, 'PNG', x, y, imgWidth, imgHeight, '',
'FAST');
            doc.save(`tarjeta_${numeroTarjeta}.pdf`);
            await actualizarContadorImpresiones(numeroTarjeta);
        } catch (error) {
            console.error('Error generando PDF:', error);
            alert('Error al generar PDF');
        }
    }

    /**
     * Imprimir directamente (botón Imprimir)
     */
    async function imprimirCodigo(numeroTarjeta, barcodeId) {
        try {
            // 1. Crear ventana popup temporal
            const printWindow = window.open('', '', 'width=700,height=400');

            // 2. Generar código de barras en canvas y convertirlo en imagen
            const canvas = generarCodigoBarrasParaImpresion(numeroTarjeta);
            const imgData = canvas.toDataURL('image/png');

            // 3. Escribir HTML de impresión con CSS de tamaño etiqueta y horizontal
            printWindow.document.write(`

```

```

<!DOCTYPE html>
<html>
<head>
    <title>Imprimir código de barras</title>
    <style>
        @media print {
            @page {
                size: 62mm 29mm;
                margin: 0;
            }
            body {
                width: 62mm;
                height: 29mm;
                margin: 0;
                padding: 0;
                box-sizing: border-box;
            }
        }
        html, body {
            width: 62mm;
            height: 29mm;
            margin: 0;
            padding: 0;
            display: flex;
            align-items: center;
            justify-content: center;
            background: #fff;
        }
        .barcode-container {
            text-align: center;
            width: 100%;
        }
        .barcode-image {
            width: 58mm;
            max-width: 100%;
            height: 20mm;
            object-fit: contain;
        }
    </style>
</head>
<body>
    <div class="barcode-container">
        
    </div>
    <script>
        window.onload = function() {
            window.print();

```

```

        window.onafterprint = function() {
window.close(); };
    };
    </script>
</body>
</html>
`);
await actualizarContadorImpresiones(numeroTarjeta);
} catch (error) {
    console.error('Error imprimiendo:', error);
    alert('Error al imprimir el código de barras');
}
}

/***
 * Actualiza el contador de impresiones en la base de datos
 */
async function actualizarContadorImpresiones(numeroTarjeta) {
try {
    await fetch("/CRM_INT/CRM/controller/CodigoController.php", {
        method: "POST",
        headers: { "Content-Type": "application/x-www-form-
urlencoded" },
        body:
`action=incrementarImpresiones&idCliente=${numeroTarjeta}`
    });
    cargarCódigos();
} catch (error) {
    console.error('Error actualizando contador:', error);
}
}

/***
 * Redirige a CompraView con el idCliente (botón invisible sobre el
código de barras)
 */
window.redirigirCompra = function(idCliente) {
    console.log('== CLICK EN CODIGO DE BARRAS ==');
    console.log('ID Cliente:', idCliente);
    console.log('Tipo de idCliente:', typeof idCliente);

    if (!idCliente) {
        console.error('Error: idCliente está vacío');
        alert('Error: Número de tarjeta no válido');
        return;
    }
}

```

```

        const url =
`/CRM_INT/CRM/index.php?view=compras&idCliente=${encodeURIComponent(idCliente)}&buscar=auto`;
        console.log('URL de redirección:', url);
        console.log('Redirigiendo...');

        window.location.href = url;
    }

/***
 * Muestra la tabla de códigos
 */
function mostrarCódigos(códigos) {
    const contenedor = document.getElementById("codigoLista");

    let filas = códigos.map((código, index) => {
        const barcodeId = `barcode-${index}`;
        return `
            <tr>
                <td>${código.idCliente}</td>
                <td>${código.cedula ?? '-'}</td>
                <td>${código.nombre ?? '-'}</td>
                <td>${código.fechaRegistro ?? '-'}</td>
                <td>
                    <div class="text-center position-relative"
style="width:120px; height:60px; margin:auto;">
                        <canvas id="${barcodeId}" style="display:block;
margin:0 auto; width:100%; height:auto;"></canvas>
                        <button
                            class="btn-barcode-invisible"
                            title="Ir a Beneficio - Tarjeta:
${código.idCliente}"
                            data-tarjeta="${código.idCliente}"
                            tabindex="-1"
                            style="position:absolute;top:0;left:0;width:100%;height:100%;opacity:0.1;cursor:pointer;border:1px solid red;padding:0;margin:0;z-index:10;background:rgba(255,0,0,0.1);">
                            </button>
                    </div>
                </td>
                <td class="text-center">${código.cantImpresiones}</td>
                <td class="text-center">
                    <div class="btn-group" role="group">
                        <button type="button" class="btn btn-sm btn-
outline-danger"
                            onclick="imprimirPDF('${código.idCliente}
', '${barcodeId}')"
                            title="Descargar PDF">

```

```

                <i class="fas fa-file-pdf"></i> PDF
            </button>
            <button type="button" class="btn btn-sm btn-
outline-success"
                    onclick="imprimirCodigo('${codigo.idClien
te}', '${barcodeId}')"
                    title="Imprimir">
                <i class="fas fa-print"></i> Imprimir
            </button>
        </div>
    </td>
</tr>
`;
}).join("");
contenedor.innerHTML = `


| Número de Tarjeta | Cédula | Nombre | Fecha Registro | Código de barras | Nº Impresiones | Imprimir |
|-------------------|--------|--------|----------------|------------------|----------------|----------|
|-------------------|--------|--------|----------------|------------------|----------------|----------|


`;
setTimeout(() => {
    codigos.forEach((codigo, index) => {
        const barcodeId = `barcode-${index}`;
        generarCodigoBarras(codigo.idCliente, barcodeId);
    });
    // Agregar event listeners a todos los botones invisibles
    const botonesInvisibles = document.querySelectorAll('.btn-
barcode-invisible');
    console.log('Configurando', botonesInvisibles.length, 'botones
invisibles');

    botonesInvisibles.forEach(boton => {
        boton.addEventListener('click', function(e) {

```

```

        e.preventDefault();
        e.stopPropagation();
        const tarjeta = this.getAttribute('data-tarjeta');
        console.log('Click en botón invisible, tarjeta:',
tarjeta);

        // Redirección directa sin llamar a función externa
        const url =
`/CRM_INT/CRM/index.php?view=compras&idCliente=${encodeURIComponent(tarje
ta)}&buscar=auto`;
        console.log('Redirigiendo directamente a:', url);
        console.log('Ejecutando window.location.href...');

        window.location.href = url;

        console.log('window.location.href ejecutado');
    });
});

// Refuerzo: deshabilitar botones de imprimir si la función
global existe
if (window.disableImprimirBtns) {
    window.disableImprimirBtns();
}
}, 100);
}

/***
 * Filtro de búsqueda en vivo
 */
document.getElementById('buscarCodigo').addEventListener('input',
function () {
    const valor = this.value.trim();
    if (!valor) {
        mostrarCodigos(codigosGlobal);
        return;
    }
    const filtrados = codigosGlobal.filter(c => {
        const tarjeta = String(c.idCliente).toLowerCase();
        const nombre = (c.nombre || '').toLowerCase();
        return tarjeta.includes(valor.toLowerCase()) ||
nombre.includes(valor.toLowerCase());
    });
    mostrarCodigos(filtrados);
});

/***
 * Función para redirigir a beneficios desde el campo de búsqueda
*/

```

```

/**
 * Función para redirigir a beneficios desde el campo de búsqueda
 */
function buscarEnBeneficios() {
    const valor = document.getElementById('buscarCodigo').value.trim();
    if (!valor) {
        alert('Por favor ingrese un número de tarjeta');
        return;
    }

    // Validar que solo contenga números
    const esNumerico = /^d+$/.test(valor);
    if (!esNumerico) {
        alert('Por favor ingrese solo números');
        return;
    }

    console.log('Redirigiendo a beneficios desde búsqueda con tarjeta:', valor);
    window.location.href =
`/CRM_INT/CRM/index.php?view=compras&idCliente=${encodeURIComponent(valor)}&buscar=auto`;
}

/**
 * Event listener para el botón de búsqueda
 */
document.getElementById('btnBuscarTarjeta').addEventListener('click', buscarEnBeneficios);

/**
 * Event listener para el botón de búsqueda
 */
document.getElementById('btnBuscarTarjeta').addEventListener('click', buscarEnBeneficios);

/**
 * Event listener para Enter en el campo de búsqueda (además del lector de códigos)
 */
document.getElementById('buscarCodigo').addEventListener('keydown', function(e) {
    if (e.key === 'Enter' && !isScanning) {
        // Si no está en modo scanning (lector), tratar como búsqueda manual
        e.preventDefault();
        buscarEnBeneficios();
    }
})

```

```

});
```

```

/***
 * Botón actualizar recarga la lista
 */
document.getElementById('btnActualizar').addEventListener('click',
cargarCódigos);
```

```
// ----- DETECCIÓN MEJORADA DE LECTOR DE CÓDIGO DE BARRAS -----
```

```
document.addEventListener("DOMContentLoaded", function() {
    cargarCódigos();

    // Variables para la detección del lector
    const inputBusqueda = document.getElementById("buscarCódigo");
    let buffer = '';
    let lastKeyTime = Date.now();
    let isScanning = false;
    let scanTimeout;

    // Función para procesar el código escaneado
    function procesarCódigoEscaneado(código) {
        console.log('Código escaneado detectado:', código);

        // Quitar ceros a la izquierda (00000081 → 81)
        const códigoLimpio = parseInt(código, 10).toString();
        console.log('Código limpio (sin ceros):', códigoLimpio);

        // Limpiar el input
        inputBusqueda.value = '';

        // Redirigir a compras con el código limpio
        const url =
`/CRM_INT/CRM/index.php?view=compras&idCliente=${encodeURIComponent(códigoLimpio)}`;
        console.log('Redirigiendo a:', url);
        window.location.href = url;
    }

    // Función para resetear el buffer
    function resetBuffer() {
        buffer = '';
        isScanning = false;
        if (scanTimeout) {
            clearTimeout(scanTimeout);
        }
    }
})
```

```

// Evento keydown mejorado
inputBusqueda.addEventListener('keydown', function(e) {
    const currentTime = Date.now();
    const timeDiff = currentTime - lastKeyTime;

    // Detectar si es entrada rápida (típico de lector de código de barras)
    if (timeDiff < 50) {
        isScanning = true;
    }

    lastKeyTime = currentTime;

    // Si pasa mucho tiempo entre teclas, resetear buffer
    if (timeDiff > 200) {
        resetBuffer();
    }

    // Manejar Enter
    if (e.key === "Enter") {
        e.preventDefault();

        if (buffer.length > 0) {
            procesarCodigoEscaneado(buffer);
            resetBuffer();
            return;
        }
    }

    // Manejar caracteres normales
    if (e.key.length === 1) {
        buffer += e.key;

        // Timeout para procesar automáticamente después de cierto tiempo
        clearTimeout(scanTimeout);
        scanTimeout = setTimeout(() => {
            if (buffer.length >= 4 && isScanning) {
                procesarCodigoEscaneado(buffer);
                resetBuffer();
            }
        }, 100);
    }
});

// Evento keyup para capturar casos donde el Enter no se detecta bien
inputBusqueda.addEventListener('keyup', function(e) {
    if (e.key === "Enter" && buffer.length > 0) {

```

```

        procesarCodigoEscaneado(buffer);
        resetBuffer();
    }
});

// Evento input como respaldo
inputBusqueda.addEventListener('input', function(e) {
    const valor = this.value.trim();

    // Si el valor cambió rápidamente y es largo, podría ser un
escaneo
    if (valor.length >= 4 && isScanning) {
        setTimeout(() => {
            if (this.value === valor) {
                procesarCodigoEscaneado(valor);
                resetBuffer();
            }
        }, 200);
    }
});

// Resetear buffer cuando se enfoca el input
inputBusqueda.addEventListener('focus', function() {
    resetBuffer();
});

// Asegurar que el input esté siempre enfocado para capturar la
entrada del lector
inputBusqueda.focus();

// Re-enfocar el input periódicamente
setInterval(() => {
    if (document.activeElement !== inputBusqueda) {
        inputBusqueda.focus();
    }
}, 1000);
});

// Función de respaldo para probar manualmente
window.testRedireccion = function(idCliente) {
    const url =
`/CRM_INT/CRM/index.php?view=compras&idCliente=${encodeURIComponent(idCli
ente)}&buscar=auto`;
    console.log('URL de prueba:', url);
    window.location.href = url;
}

```

```

// Función de debugging para verificar que las funciones están
disponibles
window.debugCódigos = function() {
    console.log('== DEBUG CÓDIGOS ==');
    console.log('redirigirCompra function:', typeof
window.redirigirCompra);
    console.log('testRedireccion function:', typeof
window.testRedireccion);
    console.log('códigosGlobal:', códigosGlobal.length, 'códigos
disponibles');

    if (códigosGlobal.length > 0) {
        console.log('Primer código:', códigosGlobal[0]);
        console.log('Para probar, ejecuta: testRedireccion("' +
códigosGlobal[0].idCliente + '")');
        console.log('O ejecuta: redirigirCompra("' +
códigosGlobal[0].idCliente + '")');
    }
    console.log('=====');
}

// Función para probar el click manualmente
window.testClick = function(idCliente) {
    console.log('== TEST MANUAL DE CLICK ==');
    if (!idCliente && códigosGlobal.length > 0) {
        idCliente = códigosGlobal[0].idCliente;
    }
    console.log('Probando con ID:', idCliente);
    redirigirCompra(idCliente);

    const inputBuscar = document.getElementById('buscarCódigo');
    const btnBuscar = document.getElementById('btnBuscarTarjeta');

    function validarBotónBuscarTarjeta() {
        const valor = inputBuscar.value.trim();
        const esNúmero = /^d+$/.test(valor); // Solo números enteros
positivos

        if (esNúmero) {
            btnBuscar.disabled = false;
            btnBuscar.classList.remove('btn-disabled');
        } else {
            btnBuscar.disabled = true;
            btnBuscar.classList.add('btn-disabled');
        }
    }
}

```

```

// Validar al escribir
inputBuscar.addEventListener('input', validarBotonBuscarTarjeta);

// Validar al cargar
validarBotonBuscarTarjeta();

}

```

## |   |— Compra.js

```

document.addEventListener("DOMContentLoaded", function () {
    // Elementos DOM
    const elements = {
        btns: document.querySelectorAll(".compra-btn-opcion"),
        btnBuscar: document.getElementById("compraBtnBuscar"),
        btnBuscarIcon: document.getElementById("compraBtnBuscarIcon"),
        btnCompra: document.getElementById("compraOpcionCompra"),
        btnDescuento: document.getElementById("compraOpcionDescuento"),
        btnExpress: document.getElementById("compraOpcionExpress"),
        inputAcumulada: document.getElementById("cantidadAcumulada"),
        inputId: document.getElementById("compraInputId"),
        compraCardForm: document.getElementById("compraCardForm"),
        compraIndicacion: document.getElementById("compraIndicacion"),
        historialCompras: document.getElementById("historialCompras"),
    };

    // Estado global
    const state = {
        idClienteActual: null,
        nombreClienteActual: null,
        datosClienteActual: null,
        expressTimer: null, // Para manejar el temporizador
        expressValidado: false, // Para saber si Express ya fue validado
    };

    // Configuración de botones
    const buttonConfig = {
        compra: {
            buscarText: "Buscar",
            acumularText: "Acumular",
            background: "var(--amarillo)",
            acumularBackground: "#39cc6b",
            color: "var(--negro)",
            acumularColor: "white",
            showIcon: false,
        }
    };
}

```

```

        showBuscar: true,
        readonly: false,
    },
    descuento: {
        buscarText: "Aplicar",
        background: "#198754",
        color: "white",
        showIcon: false,
        showBuscar: true,
        readonly: true,
    },
    express: {
        buscarText: "Buscar",
        background: "var(--amarillo)",
        color: "var(--negro)",
        showIcon: false,
        showBuscar: false,
        readonly: true,
    },
};

// Inicialización
init();

function init() {
    handleURLParams();
    setupEventListeners();
    // El botón Express inicia oculto
    if (elements.btnExpress) {
        elements.btnExpress.style.display = "none";
    }
}

function handleURLParams() {
    const params = new URLSearchParams(window.location.search);
    const idAuto = params.get("idCliente");
    const buscarAuto = params.get("buscar");

    console.log("Parámetros URL detectados:", { idAuto, buscarAuto });

    if (idAuto) {
        console.log("Configurando búsqueda automática para tarjeta:",
        idAuto);

        selectButton("compra");
        elements.inputId.value = idAuto;

        if (buscarAuto === "auto") {

```

```

        console.log("Ejecutando búsqueda automática...");
        setTimeout(() => {
            elements.btnBuscar.click();
        }, 300);
    }
}

function setupEventListeners() {
    // Enter en input
    elements.inputId.addEventListener("keydown", function (e) {
        if (e.key === "Enter") {
            selectButton("compra");
            e.preventDefault();
            elements.btnBuscar.click();
        }
    });
}

// Botones de opciones
elements.btns.forEach((btn) => {
    btn.addEventListener("click", function () {
        const buttonType = getButtonType(this);

        // Si es Express y ya está seleccionado, deseleccionar y volver
        // al estado inicial
        if (buttonType === "express" &&
this.classList.contains("selected")) {
            deselectAllButtons();
            resetToInitialState();
            return;
        }

        selectButton(buttonType);

        // Si es Express y hay cliente, mostrar modal inmediatamente
        if (
            buttonType === "express" &&
            state.idClienteActual &&
            state.datosClienteActual
        ) {
            mostrarModalExpress();
        }
    });
});

// Eventos de búsqueda
elements.btnBuscar.addEventListener("click", manejarAccion);
elements.btnBuscarIcon.addEventListener("click", buscarCliente);

```

```

}

function getButtonType(button) {
    if (button === elements.btnCompra) return "compra";
    if (button === elements.btnDescuento) return "descuento";
    if (button === elements.btnExpress) return "express";
    return null;
}

function selectButton(type) {
    // Si Express ya está validado, no deseleccionarlo nunca
    if (state.expressValidado) {
        // Solo deseleccionar los otros botones (no Express)
        elements.btns.forEach((b) => {
            if (getButtonType(b) !== "express") {
                b.classList.remove("selected");
            }
        });
    }

    // Asegurar que Express se mantenga seleccionado
    elements.btnExpress.classList.add("selected");
} else {
    deselectAllButtons();
}

const button =
    type === "compra"
        ? elements.btnCompra
        : type === "descuento"
        ? elements.btnDescuento
        : elements.btnExpress;

if (button) {
    button.classList.add("selected");
}

updateButtonState(type);
}

function deselectAllButtons() {
    elements.btns.forEach((b) => b.classList.remove("selected"));
}

function resetToInitialState() {
    // Limpiar estado del cliente
    state.idClienteActual = null;
    state.nombreClienteActual = null;
    state.datosClienteActual = null;
}

```

```

state.expressValidado = false; // Resetear estado Express

// Limpiar temporizador si existe
if (state.expressTimer) {
    clearInterval(state.expressTimer);
    state.expressTimer = null;
}

// Limpiar formularios
elements.inputId.value = "";
elements.inputAcumulada.value = "";

// Ocultar formulario del cliente y mostrar indicación inicial
elements.compraCardForm.style.display = "none";
elements.compraIndicacion.style.display = "block";

// Ocultar botón Express nuevamente
if (elements.btnExpress) {
    elements.btnExpress.style.display = "none";
}

// Restablecer botón buscar al estado inicial
elements.btnBuscar.style.display = "inline-block";
elements.btnBuscar.textContent = "Buscar";
elements.btnBuscar.style.background = "var(--amarillo)";
elements.btnBuscar.style.color = "var(--negro)";
elements.btnBuscarIcon.style.display = "none";

// Restablecer input acumulada
elements.inputAcumulada.removeAttribute("readonly");

// Limpiar historial
if (elements.historialCompras) {
    elements.historialCompras.innerHTML = "";
}
}

function updateButtonState(type) {
    const config = buttonConfig[type];
    if (!config) return;

    // Configurar readonly del input
    if (config.readonly) {
        elements.inputAcumulada.setAttribute("readonly", true);
    } else {
        elements.inputAcumulada.removeAttribute("readonly");
    }
}

```

```

// Mostrar/ocultar botón buscar según configuración
elements.btnBuscar.style.display = config.showBuscar
    ? "inline-block"
    : "none";

// Solo actualizar botón si es visible
if (config.showBuscar) {
    // Para compra, verificar si hay cliente para mostrar "Acumular"
    if (type === "compra" && state.idClienteActual) {
        elements.btnBuscar.textContent = config.acumularText;
        elements.btnBuscar.style.background = config.acumularBackground;
        elements.btnBuscar.style.color = config.acumularColor;
        elements.btnBuscarIcon.style.display = "block";
    } else if (type === "descuento" && state.expressValidado) {
        // Cambiar texto del botón cuando Express está validado
        elements.btnBuscar.textContent = "Aplicar";
        elements.btnBuscar.style.background = "#28a745"; // Verde más brillante
        elements.btnBuscar.style.color = "white";
        elements.btnBuscarIcon.style.display = config.showIcon ? "block"
    : "none";
    } else {
        elements.btnBuscar.textContent = config.buscarText;
        elements.btnBuscar.style.background = config.background;
        elements.btnBuscar.style.color = config.color;
        elements.btnBuscarIcon.style.display = config.showIcon
            ? "block"
            : "none";
    }
}

function getSelectedOption() {
    const opciones = [];
    if (elements.btnCompra.classList.contains("selected"))
opciones.push("compra");
    if (elements.btnDescuento.classList.contains("selected"))
opciones.push("descuento");
    if (elements.btnExpress.classList.contains("selected"))
opciones.push("express");

    // Para compatibilidad con código existente, devolver la primera
    opción no-express
    // o express si es la única seleccionada
    const nonExpress = opciones.find(op => op !== "express");
    return nonExpress || (opciones.includes("express") ? "express" :
null);
}

```

```

async function manejarAccion() {
  const inputId = elements.inputId.value.trim();
  const opcion = getSelectedOption();

  if (!opcion || !inputId) {
    return alert("Seleccione una opción y digite la tarjeta.");
  }

  // Manejo específico por opción
  switch (opcion) {
    case "descuento":
      await handleDescuento();
      break;
    case "compra":
      if (elements.btnBuscar.textContent === "Acumular") {
        await handleCompra();
        return;
      }
      break;
    case "express":
      // Para Express, solo buscar cliente y mostrar modal
      break;
  }

  await buscarCliente();

  // Actualizar estado después de la búsqueda
  if (opcion === "compra" && state.idClienteActual) {
    updateButtonState("compra");
  } else if (
    opcion === "express" &&
    state.idClienteActual &&
    state.datosClienteActual
  ) {
    mostrarModalExpress();
  }
}

async function handleDescuento() {
  const isExpressMode = state.expressValidado &&
  elements.btnExpress.classList.contains("selected");
  const descuentoAplicar = isExpressMode ? 20000 : 50000;
  const nombreDescuento = isExpressMode ? "Express" : "VIP";

  if (elements.btnBuscar.textContent !== "Aplicar" &&
  elements.btnBuscar.textContent !== "Aplicar Express") return;
}

```

```

let saldo = parseFloat(document.getElementById("totalActual").value)
|| 0;

if (saldo < descuentoAplicar) {
    return alert(
        `El saldo actual (\${saldo.toLocaleString("es-CR")}) es
insuficiente para aplicar el descuento \${nombreDescuento} de
\${descuentoAplicar.toLocaleString("es-CR")}.`);
}

let saldoFinal = saldo - descuentoAplicar;

try {
    const response = await fetch(
        "/CRM_INT/CRM/controller/ClienteController.php",
        {
            method: "POST",
            headers: { "Content-Type": "application/x-www-form-urlencoded" },
            body: buildUpdateClienteBody(saldoFinal),
        }
    );

    const json = await response.json();
    if (!json.success) {
        return alert(
            "Ocurrió un error al actualizar el saldo en la base de datos."
        );
    }
}

document.getElementById("totalActual").value = saldoFinal;
state.datosClienteActual.acumulado = saldoFinal;

// Mensaje personalizado según el tipo de descuento
let mensaje = "";
if (isExpressMode) {
    mensaje = `¡Descuento Express aplicado exitosamente!\n\n` +
        `✓ Se descontaron
\${descuentoAplicar.toLocaleString("es-CR")}\ de tu saldo\n` +
        `⌚ Saldo anterior: \${saldo.toLocaleString("es-
CR")}\n` +
        `⌚ Saldo actual: \${saldoFinal.toLocaleString("es-
CR")}\n\n` +
        `🎉 ¡El Express ahora es GRATIS para
\${state.nombreClienteActual}!`;
}
else {
}

```

```

        mensaje = `Descuento VIP exitoso, puedes aplicar el 15% a nombre
del cliente VIP: ${state.nombreClienteActual}.\n` +
                    `El saldo actual cambió de ${saldo.toLocaleString("es-
CR")}` + `${saldoFinal.toLocaleString("es-CR")}.`;
    }

    alert(mensaje);
} catch {
    alert("Error de servidor.");
}
}

async function handleCompra() {
    const monto = parseFloat(elements.inputAcumulada.value.trim()) || 0;
    if (monto <= 0) {
        return alert("Ingrese un monto válido para acumular.");
    }

    try {
        const response = await fetch(
            "/CRM_INT/CRM/controller/CompraController.php",
            {
                method: "POST",
                headers: { "Content-Type": "application/x-www-form-urlencoded" },
                body:
`action=create&total=${monto}&idCliente=${state.idClienteActual}`,
            }
        );

        const resCompra = await response.json();
        if (resCompra.success) {
            elements.inputAcumulada.value = "";
            await buscarCliente();
            cargarHistorialCompras(state.idClienteActual);
            alert(resCompra.message);
        }
    } catch {
        alert("Error de servidor al registrar la compra.");
    }
}

function buildUpdateClienteBody(saldoFinal) {
    if (!state.datosClienteActual) return "";

    const data = state.datosClienteActual;
    return (

```

```

`action=update&id=${data.id}&cedula=${data.cedula}&nOMBRE=${data.no
mbre}` +
`&correo=${data.correo}&telefono=${data.telefono}&lugarResidencia=$
{data.lugarResidencia}` +
`&fechaCumpleanos=${data.fechaCumpleanos}&acumulado=${saldoFinal}`
);
}

async function buscarCliente() {
  const inputId = elements.inputId.value.trim();
  console.log("Valor que se envía al servidor:", inputId);

  try {
    const response = await fetch(
      "/CRM_INT/CRM/controller/ClienteController.php",
      {
        method: "POST",
        headers: { "Content-Type": "application/x-www-form-urlencoded" }
      },
      body: `action=read&id=${encodeURIComponent(inputId)}`,
    );
  }

  const { success, data } = await response.json();
  if (!success || !data) {
    return alert("No se encontró un cliente con ese número de
tarjeta.");
  }

  updateClienteState(data);
  populateClienteForm(data);
  showClienteForm();

  // Mostrar el botón Express después de la primera búsqueda exitosa
  if (elements.btnExpress) {
    elements.btnExpress.style.display = "inline-block";
  }

  cargarHistorialCompras(state.idClienteActual);
  handleBirthdayConfetti(data);
} catch {
  alert("Error de conexión con el servidor.");
}
}

function updateClienteState(data) {
  state.datosClienteActual = { ...data, acumulado: data.acumulado || 0
};

```

```

        state.idClienteActual = data.id;
        state.nombreClienteActual = data.nombre;
    }

    function populateClienteForm(data) {
        const fields = [
            { id: "id", value: data.id },
            { id: "clienteCedula", value: data.cedula },
            { id: "clienteNombre", value: data.nombre },
            { id: "clienteCorreo", value: data.correo },
            { id: "clienteTelefono", value: data.telefono },
            { id: "clienteLugar", value: data.lugarResidencia },
            { id: "clienteFecha", value: data.fechaCumpleanos },
            { id: "totalActual", value: state.datosClienteActual.acumulado },
        ];

        fields.forEach((field) => {
            const element = document.getElementById(field.id);
            if (element) {
                element.value = field.value || "";
            }
        });
    }

    elements.inputAcumulada.value = "";
}

function showClienteForm() {
    elements.compraCardForm.style.display = "block";
    elements.compraIndicacion.style.display = "none";
}

function handleBirthdayConfetti(data) {
    if (!data.fechaCumpleanos || typeof confetti !== "function") return;

    if (esCumpleEstaSemana(data.fechaCumpleanos)) {
        // Múltiples efectos de confeti
        const confettiEffects = [
            { particleCount: 100, spread: 120, origin: { y: 0.6, x: 0.5 } },
            { particleCount: 80, spread: 120, origin: { y: 0.6, x: 0.1 } },
            { particleCount: 80, spread: 120, origin: { y: 0.6, x: 0.9 } },
        ];

        confettiEffects.forEach((effect) => confetti(effect));

        setTimeout(() => {
            mostrarModalCumpleanos(data.nombre);
        }, 1000);
    }
}

```

```

}

function esCumpleEstaSemana(fechaCumpleanos) {
  const hoy = new Date();
  const diaSemana = hoy.getDay() === 0 ? 7 : hoy.getDay();

  const lunes = new Date(hoy);
  lunes.setHours(0, 0, 0, 0);
  lunes.setDate(hoy.getDate() - (diaSemana - 1));

  const domingo = new Date(lunes);
  domingo.setDate(lunes.getDate() + 6);
  domingo.setHours(23, 59, 59, 999);

  const cumple = new Date(fechaCumpleanos);
  cumple.setFullYear(hoy.getFullYear());

  return cumple >= lunes && cumple <= domingo;
}

function mostrarModalExpress() {
  const modal = document.getElementById("modalExpress");
  if (!modal) {
    console.log("Modal Express no encontrado en la vista");
    return;
  }

  // Resetear el modal al estado inicial
  resetModalExpress();

  // Actualizar datos del cliente en el modal
  updateModalData("expressNombre", state.datosClienteActual.nombre);
  updateModalData("expressCorreo", state.datosClienteActual.correo);

  // Configurar botón de envío
  const btnEnviar = document.getElementById("btnEnviarCorreoExpress");
  if (btnEnviar) {
    btnEnviar.onclick = enviarCorreoExpress;
  }

  // Mostrar modal
  const modalInstance = new bootstrap.Modal(modal);
  modalInstance.show();
}

function resetModalExpress() {
  // Limpiar temporizador anterior si existe
  if (state.expressTimer) {

```

```

        clearInterval(state.expressTimer);
        state.expressTimer = null;
    }

    // Restaurar contenido inicial del modal
    const modalBody = document.querySelector('#modalExpress .modal-body');
    if (modalBody) {
        modalBody.innerHTML =
            `

Se enviará un código de verificación al correo:


<strong id="expressCorreo"></strong></p>
            <p class="text-muted">El código expirará en 5 minutos.</p>
        `;
    }

    // Restaurar botón del footer
    const btnEnviar = document.getElementById("btnEnviarCorreoExpress");
    if (btnEnviar) {
        btnEnviar.textContent = "Enviar Correo";
        btnEnviar.disabled = false;
        btnEnviar.className = "btn btn-primary";
    }
}

async function enviarCorreoExpress() {
    console.log("Enviando correo Express...");

    if (!state.idClienteActual || !state.datosClienteActual) {
        alert("Error: No hay cliente seleccionado");
        return;
    }

    try {
        // Deshabilitar botón y mostrar estado de carga
        const btnEnviar =
document.getElementById("btnEnviarCorreoExpress");
        btnEnviar.disabled = true;
        btnEnviar.textContent = "Enviando...";

        // 1) Generar y enviar código
        const body =
`action=generarCodigoExpress&idCliente=${state.idClienteActual} +
    &correo=${encodeURIComponent(state.datosClienteActual
.correo)} +
    &nombre=${encodeURIComponent(state.datosClienteActual
.nombre)}`;
    
```

```

        const response = await
fetch('/CRM_INT/CRM/controller/ExpressController.php', {
    method: 'POST',
    headers: {'Content-Type': 'application/x-www-form-urlencoded'},
    body: body
});

const json = await response.json();

if (!json.success) {
    btnEnviar.disabled = false;
    btnEnviar.textContent = "Enviar Correo";
    alert('Error al enviar correo Express: ' + (json.message ||
'Error desconocido'));
    return;
}

console.log("Correo enviado exitosamente, cambiando modal...");

// 2) Cambiar modal a modo validación
cambiarModalAValidacion();

} catch (error) {
    console.error("Error al enviar correo:", error);
    const btnEnviar =
document.getElementById("btnEnviarCorreoExpress");
    btnEnviar.disabled = false;
    btnEnviar.textContent = "Enviar Correo";
    alert('Error de conexión al enviar el correo');
}
}

function cambiarModalAValidacion() {
    const modalBody = document.querySelector('#modalExpress .modal-
body');
    if (!modalBody) return;

    modalBody.innerHTML = `
<div class="text-center mb-3">
    <p class="mb-2">Código enviado a:
<strong>${state.datosClienteActual.correo}</strong></p>
    <div class="h4 mb-3 text-primary" id="relojExpress">05:00</div>
</div>
<div class="input-group mb-3">
    <input type="number" class="form-control" id="inputCodigoExpress"
placeholder="Ingresa el código de 6 dígitos"
maxlength="6">

```

```

        <button class="btn btn-success"
id="btnValidarExpress">Validar</button>
    </div>
    <div id="msgExpress" class="text-danger text-center"></div>
`;

// Cambiar el botón del footer
const btnEnviar = document.getElementById("btnEnviarCorreoExpress");
if (btnEnviar) {
    btnEnviar.textContent = "Reenviar Código";
    btnEnviar.disabled = false;
    btnEnviar.className = "btn btn-outline-secondary";
    btnEnviar.onclick = enviarCorreoExpress; // Permitir reenvío
}

// 3) Iniciar temporizador de 5 minutos
iniciarTemporizador();

// 4) Configurar validación
document.getElementById('btnValidarExpress').onclick =
validarCodigoExpress;

// Auto-focus en el input
document.getElementById('inputCodigoExpress').focus();
}

function iniciarTemporizador() {
    let restantes = 300; // 5 minutos en segundos
    const reloj = document.getElementById('relojExpress');

    if (!reloj) return;

    // Limpiar temporizador anterior si existe
    if (state.expressTimer) {
        clearInterval(state.expressTimer);
    }

    state.expressTimer = setInterval(() => {
        restantes--;
        const min = String(Math.floor(restantes / 60)).padStart(2, '0');
        const seg = String(restantes % 60).padStart(2, '0');
        reloj.textContent = `${min}:${seg}`;

        if (restantes <= 0) {
            clearInterval(state.expressTimer);
            state.expressTimer = null;
            reloj.textContent = '00:00';
            reloj.className = 'h4 mb-3 text-danger';
        }
    }, 1000);
}

```

```

        const msgExpress = document.getElementById('msgExpress');
        if (msgExpress) {
            msgExpress.textContent = 'Código expirado. Solicita un nuevo
código.';
        }

        // Deshabilitar botón de validar
        const btnValidar = document.getElementById('btnValidarExpress');
        if (btnValidar) {
            btnValidar.disabled = true;
        }
    }
}, 1000);
}

async function validarCodigoExpress() {
    const inputCodigo = document.getElementById('inputCodigoExpress');
    const codigo = inputCodigo.value.trim();
    const msgExpress = document.getElementById('msgExpress');

    if (!codigo) {
        msgExpress.textContent = 'Por favor, digite el código';
        inputCodigo.focus();
        return;
    }

    if (codigo.length !== 6) {
        msgExpress.textContent = 'El código debe tener 6 dígitos';
        inputCodigo.focus();
        return;
    }

    try {
        const btnValidar = document.getElementById('btnValidarExpress');
        btnValidar.disabled = true;
        btnValidar.textContent = 'Validando...';

        const response = await
fetch('/CRM_INT/CRM/controller/ExpressController.php', {
            method: 'POST',
            headers: {'Content-Type': 'application/x-www-form-urlencoded'},
            body:
`action=validarCodigoExpress&idCliente=${state.idClienteActual}&codigo=${
codigo}`
        });
    }

    const json = await response.json();
}

```

```

if (json.success) {
    // Código validado correctamente
    clearInterval(state.expressTimer);
    state.expressTimer = null;

    // MARCAR EXPRESS COMO VALIDADO
    state.expressValidado = true;

    // Cerrar modal
    const modal =
bootstrap.Modal.getInstance(document.getElementById('modalExpress'));
    if (modal) {
        modal.hide();
    }

    // Mostrar confirmación
    alert('¡Código validado correctamente! Express activado.');

    // MANTENER EXPRESS SELECCIONADO y actualizar estado visual
    elements.btnExpress.classList.add("selected");

    // Agregar indicador visual de Express validado
    if (!elements.btnExpress.querySelector('.express-validado')) {
        const indicator = document.createElement('span');
        indicator.className = 'express-validado';
        indicator.innerHTML = ' ✓';
        indicator.style.color = '#28a745';
        indicator.style.fontWeight = 'bold';
        elements.btnExpress.appendChild(indicator);
    }

    // Si hay otra opción seleccionada, actualizar su estado
    const otherSelected = getSelectedOption();
    if (otherSelected && otherSelected !== "express") {
        updateButtonState(otherSelected);
    }

    console.log("Express validado y persistente para combinar con
otras opciones");

} else {
    msgExpress.textContent = json.message || 'Código incorrecto';
    btnValidar.disabled = false;
    btnValidar.textContent = 'Validar';
    inputCodigo.focus();
    inputCodigo.select();
}

```

```

    } catch (error) {
      console.error("Error al validar código:", error);
      msgExpress.textContent = 'Error de conexión al validar el código';
      const btnValidar = document.getElementById('btnValidarExpress');
      btnValidar.disabled = false;
      btnValidar.textContent = 'Validar';
    }
  }

  function updateModalData(elementId, value) {
    const element = document.getElementById(elementId);
    if (element) {
      element.textContent = value || "";
    }
  }

  // Funciones globales (mantenidas para compatibilidad)
  window.cargarHistorialCompras = async function (idCliente) {
    elements.historialCompras.innerHTML =
      '<div style="color:var(--gris)">Cargando historial...</div>';

    try {
      const response = await fetch(
        "/CRM_INT/CRM/controller/CompraController.php",
        {
          method: "POST",
          headers: { "Content-Type": "application/x-www-form-urlencoded" },
          body: `action=readByCliente&idCliente=${encodeURIComponent(
            idCliente
          )}`,
        }
      );
      const json = await response.json();

      if (!json.success || !json.data || !json.data.length) {
        elements.historialCompras.innerHTML = `<div class="alert alert-info mt-4">No hay compras registradas para este cliente.</div>`;
        return;
      }

      const filas = json.data
        .slice()
        .reverse()
        .map((compra, i, arr) => buildCompraRow(compra, arr.length - i))
        .join("");
    }
  };
}

```

```

elements.historialCompras.innerHTML = `

<div class="tabla-scroll">
    <table class="table table-striped table-hover">
        <thead class="table-dark">
            <tr><th>#</th><th>Fecha</th><th>Total</th><th>Nombre
Cliente</th><th>Acciones</th></tr>
        </thead>
        <tbody>${filas}</tbody>
    </table>
</div>
`;
} catch {
    elements.historialCompras.innerHTML = `<div
style="color:#d43b3b;">Error al cargar historial</div>`;
}
};

function buildCompraRow(compra, numero) {
    return `
<tr>
    <td>${numero}</td>
    <td>${compra.fechaCompra}</td>
    <td>${parseFloat(compra.total).toLocaleString("es-CR")}</td>
    <td>${state.nombreClienteActual || ""}</td>
    <td>
        <button class="btn btn-sm btn-danger" title="Eliminar"
onclick="eliminarCompra(${compra.idCompra
})">
            <i class="bi bi-trash"></i>
        </button>
    </td>
</tr>
`;
}
}

window.eliminarCompra = async function (idCompra) {
    if (!confirm("¿Está seguro de eliminar esta compra?")) return;

    try {
        const response = await fetch(
            "/CRM_INT/CRM/controller/CompraController.php",
            {
                method: "POST",
                headers: { "Content-Type": "application/x-www-form-urlencoded" },
                body: `action=delete&idCompra=${encodeURIComponent(idCompra)}`
            }
        );
    }
}

```

```

);

const json = await response.json();
alert(json.message);

if (json.success && state.idClienteActual) {
    cargarHistorialCompras(state.idClienteActual);
}
} catch {
    alert("Error al eliminar la compra.");
}
};

// Función global para modal de cumpleaños
window.mostrarModalCumpleanos = function (nombreCliente) {
    const modal = document.getElementById("modalCumpleanos");
    if (!modal) {
        console.log("Modal de cumpleaños no encontrado en esta vista");
        return;
    }

    updateModalData("nombreCumpleanero", nombreCliente);

    const modalInstance = new bootstrap.Modal(modal);
    modalInstance.show();

    const btnIrCumpleanos = document.getElementById("btnIrCumpleanos");
    if (btnIrCumpleanos) {
        btnIrCumpleanos.onclick = function () {
            window.location.href = "/CRM_INT/CRM/index.php?view=cumple";
        };
    }
};

// Limpiar temporizador cuando se cierra el modal Express
const modalExpress = document.getElementById("modalExpress");
if (modalExpress) {
    modalExpress.addEventListener('hidden.bs.modal', function () {
        if (state.expressTimer) {
            clearInterval(state.expressTimer);
            state.expressTimer = null;
        }
    });
}

});


```

|   |— Cumple.js

```

document.addEventListener("DOMContentLoaded", () => {
    mostrarSemanaActual(); // ↪ NUEVO: Mostrar el rango apenas cargue
    cargarCumples();
    cargarHistorial();
    // --- Navegación de semana (flechas del header)
    const btnPrev = document.getElementById("btnPrevSemana");
    const btnNext = document.getElementById("btnNextSemana");
    const lblSemana = document.getElementById("lblSemana");

    if (btnPrev && btnNext) {
        btnPrev.addEventListener("click", () => {
            semanaOffset = 0;
            mostrarSemanaActual(); // repinta rango
            cargarCumples(); // recarga datos
            actualizarNavSemanaUI();
        });
        btnNext.addEventListener("click", () => {
            semanaOffset = 1;
            mostrarSemanaActual();
            cargarCumples();
            actualizarNavSemanaUI();
        });
        actualizarNavSemanaUI(); // estado inicial (< desabilitado)
    }

    document.getElementById("btnEnviarCorreo").disabled = true;
    // Botón WhatsApp inicialmente desabilitado
    const btnWhats = document.getElementById("btnEnviarWhats");
    if (btnWhats) btnWhats.disabled = true;

    // Función para actualizar el badge de cumpleaños pendientes en el
    sidebar
    function actualizarCumpleBadgeSidebar() {
        fetch("/CRM_INT/CRM/controller/CumpleController.php", {
            method: "POST",
            headers: { "Content-Type": "application/x-www-form-urlencoded" },
            body: "action=hayPendientes",
        })
        .then((res) => res.json())
        .then((data) => {
            if (window.mostrarCumpleBadge) {
                window.mostrarCumpleBadge(data.success && data.hayPendientes);
            }
        });
    }

    document
        .getElementById("FormCorreo")

```

```

.addListener("submit", async (e) => {
  e.preventDefault();

  const id = document.getElementById("idCumple").value;
  const nombre = document.getElementById("nombreCorreo").value;
  const correo = document.getElementById("correoCorreo").value;

  if (!correo || correo.trim() === "") {
    Swal.fire({
      icon: "info",
      title: "¡No tiene correo!",
      text: `${nombre} no tiene correo registrado. Contactalo por teléfono para felicitarlo.`,
    });
    return;
  }

  if (!id || !nombre || !correo) return;

  const mensaje = `¡Feliz cumpleaños ${nombre}! 🎉 En nuestro restaurante queremos celebrarte con una comida, bebida o postre totalmente gratis para vos. Te esperamos hoy mismo para consentirte como te merecés.`;

  const formData = new URLSearchParams();
  formData.append("action", "enviarCorreoCumple");
  formData.append("nombre", nombre);
  formData.append("correo", correo);
  formData.append("mensaje", mensaje);
  // NUEVO: enviar cedula, telefono y cumpleaños
  // NUEVO: enviar solo el ID del cliente
  const idCliente = document.getElementById("idCumple").value;
  formData.append("idCliente", idCliente);

  try {
    const res = await fetch(
      "/CRM_INT/CRM/controller/CumpleController.php",
      {
        method: "POST",
        headers: { "Content-Type": "application/x-www-form-urlencoded" },
        body: formData.toString(),
      }
    );

    const data = await res.json();

    if (data.success) {

```

```

        await cambiarEstado(id, "LISTA");
        Swal.fire("¡Éxito!", data.message, "success");
        document.getElementById("formCorreo").reset();
        document.getElementById("btnEnviarCorreo").disabled = true;
        cargarCumples();
        cargarHistorial();
        if (window.actualizarCumpleBadgeSidebar)
            window.actualizarCumpleBadgeSidebar();
    } else {
        Swal.fire("Error", data.message, "error");
    }
} catch (err) {
    console.error("Error al enviar correo:", err);
    Swal.fire("Error", "No se pudo conectar con el servidor",
"error");
}
});

// Handler Enviar WhatsApp
if (btnWhats) {
    btnWhats.addEventListener("click", async () => {
        const id = document.getElementById("idCumple").value;
        const nombre = document.getElementById("nombreCorreo").value;
        let telefono = document.getElementById("telefonoCorreo").value;

        if (!telefono || telefono.trim() === "") {
            Swal.fire({
                icon: "info",
                title: "Sin teléfono",
                text: `${nombre} no tiene teléfono registrado.`,
            });
            return;
        }
        if (!id || !nombre || !telefono) return;

        // Normalizar: quitar no dígitos y prefijar 506 si parece local (8
dígitos)
        const soloDigitos = (telefono || "").replace(/\D+/g, "");
        if (soloDigitos.length === 8) {
            telefono = "506" + soloDigitos;
        } else if (soloDigitos.length > 8) {
            telefono = soloDigitos; // ya tiene prefijo
        } else {
            telefono = soloDigitos; // dejar lo que haya para que backend
valide
        }
    });
}

```

```

const mensaje = `¡Hola ${nombre}! En Bastos sabemos que estás de
cumpleaños. Visítanos para celebrarlo juntos y reclamar tu regalía
& `;

const formData = new URLSearchParams();
formData.append("action", "enviarWhatsCumple");
formData.append("nombre", nombre);
formData.append("telefono", telefono);
formData.append("mensaje", mensaje);
formData.append("idCliente", id);

try {
  btnWhats.disabled = true;
  btnWhats.innerText = "Enviando...";
  const res = await fetch(
    "/CRM_INT/CRM/controller/CumpleController.php",
    {
      method: "POST",
      headers: { "Content-Type": "application/x-www-form-
urlencoded" },
      body: formData.toString(),
    }
  );
  const text = await res.text();
  let data;
  try {
    data = JSON.parse(text);
  } catch (e) {
    console.error("Respuesta no JSON:", text);
    throw new Error("Respuesta no válida del servidor");
  }

  if (data.success) {
    await cambiarEstado(id, "LISTA");
    Swal.fire("¡Éxito!", data.message, "success");
    document.getElementById("formCorreo").reset();
    document.getElementById("btnEnviarCorreo").disabled = true;
    if (btnWhats) btnWhats.disabled = true;
    cargarCumples();
    cargarHistorial();
    if (window.actualizarCumpleBadgeSidebar)
      window.actualizarCumpleBadgeSidebar();
  } else {
    Swal.fire("Error", data.message, "error");
  }
} catch (err) {
  console.error("Error al enviar WhatsApp:", err);
  Swal.fire(

```

```

        "Error",
        err && err.message
        ? err.message
        : "No se pudo conectar con el servidor",
        "error"
    );
} finally {
    btnWhats.innerHTML = '<i class="bi bi-whatsapp"></i> Enviar
WhatsApp';
    btnWhats.disabled = false;
}
});
}
});
});

let semanaOffset = 0; // 0 = actual, 1 = siguiente

const mostrarSemanaActual = () => {
// Hoy
const hoy = new Date();
const diaActual = hoy.getDay(); // 0 (Dom) a 6 (Sáb)

// Mover a lunes de la semana actual
const diffInicio = diaActual === 0 ? -6 : 1 - diaActual;
const lunes = new Date(
    hoy.getFullYear(),
    hoy.getMonth(),
    hoy.getDate() + diffInicio + semanaOffset * 7
);
const domingo = new Date(
    lunes.getFullYear(),
    lunes.getMonth(),
    lunes.getDate() + 6
);

const opciones = { day: "2-digit", month: "long" };
const formatoLunes = lunes.toLocaleDateString("es-CR", opciones);
const formatoDomingo = domingo.toLocaleDateString("es-CR", opciones);

// Rango en la franja azul
const div = document.getElementById("rangoSemana");
if (div) {
    const etiqueta = semanaOffset === 0 ? "Semana actual" : "Semana
siguiente";
    div.innerHTML = `📅 ${etiqueta}: <strong>${formatoLunes}</strong> al
<strong>${formatoDomingo}</strong>`;
}
}

```

```

// Badge del header (por si no querés usar actualizarNavSemanaUI aquí)
const lbl = document.getElementById("lblSemana");
if (lbl)
    lbl.textContent = semanaOffset === 0 ? "SEMANA ACTUAL" : "SEMANA SIGUIENTE";
};

function actualizarNavSemanaUI() {
    const btnPrev = document.getElementById("btnPrevSemana");
    const btnNext = document.getElementById("btnNextSemana");
    const lblSemana = document.getElementById("lblSemana");

    if (btnPrev) btnPrev.disabled = semanaOffset === 0;
    if (btnNext) btnNext.disabled = semanaOffset === 1;
    if (lblSemana)
        lblSemana.textContent =
            semanaOffset === 0 ? "SEMANA ACTUAL" : "SEMANA SIGUIENTE";
}

const cargarCumpleas = async () => {
    const contenedor = document.getElementById("cumpleLista");
    contenedor.innerHTML =
        `<div class="text-center p-3">
            <div class="spinner-border text-warning" role="status"></div>
            <p class="mt-2">Cargando cumpleaños...</p>
        </div>
`;
    try {
        // 🔍 ENVÍA EL OFFSET (0 = actual, 1 = siguiente)
        const body =
`action=readSemana&offset=${encodeURIComponent(semanaOffset)}`;

        const res = await
fetch("/CRM_INT/CRM/controller/CumpleController.php", {
            method: "POST",
            headers: { "Content-Type": "application/x-www-form-urlencoded" },
            body,
        });

        const data = await res.json();

        if (data.success) {
            renderizarTabla(data.data);
            // repinta el rango y actualiza flechas/badge
            mostrarSemanaActual();
            if (typeof actualizarNavSemanaUI === "function")
                actualizarNavSemanaUI();
        }
    }
}

```

```

    } else {
        contenedor.innerHTML = `<div class="alert alert-danger text-
center">${
            data.message || "No se pudo cargar."
        }</div>`;
    }
} catch (error) {
    console.error("Error cargando cumpleaños:", error);
    contenedor.innerHTML = `<div class="alert alert-danger text-
center">Error al cargar los cumpleaños.</div>`;
}
};

const renderizarTabla = (cumpleas) => {
    const contenedor = document.getElementById("cumpleLista");

    const pendientes = cumpleas.filter((c) => c.estado === "Activo");
    pendientes.sort((a, b) => {
        const fechaA = new Date(a.fechaCumpleanos);
        const fechaB = new Date(b.fechaCumpleanos);
        return fechaB - fechaA;
    });

    if (pendientes.length === 0) {
        contenedor.innerHTML = `<div class="alert alert-info text-center">No
hay cumpleaños pendientes esta semana.</div>`;
        return;
    }

    let html = `
        <div class="table-responsive">
            <table class="table table-bordered table-hover tabla-
ajustada">
                <thead>
                    <tr>
                        <th>Nombre</th>
                        <th>Cédula</th>
                        <th>Correo</th>
                        <th>Teléfono</th>
                        <th>Fecha de Cumpleaños</th>
                        <th>Estado</th>
                        <th>Acción</th>
                    </tr>
                </thead>
                <tbody>
`;
}

let recordatorios = "";

```

```

let tieneSinCorreo = false;

pendientes.forEach((c) => {
  let badge = `<span class="badge bg-success">Activo</span>`;
  if (c.estado === "LISTA") {
    badge = `<span class="badge bg-success">LISTO</span>`;
  }

  const botonCorreo = `
    <button class="btn btn-sm btn-primary"
      onclick="seleccionarCumple('${c.id}', '${c.nombre}',
      '${c.cedula}', '${c.correo}', '${c.telefono}', '${c.fechaCumpleanos}')
      <i class="fas fa-paper-plane"></i>
    </button>
  `;

  html += `
    <tr id="fila-${c.id}">
      <td>${c.nombre}</td>
      <td>${c.cedula}</td>
      <td class="correo">${c.correo}</td>
      <td>${c.telefono}</td>
      <td>${formatearFecha(c.fechaCumpleanos)}</td>
      <td>${badge}</td>
      <td class="text-center">${botonCorreo}</td>
    </tr>
  `;

  if (!c.correo || c.correo.trim() === "") {
    tieneSinCorreo = true;
    recordatorios += `
      <li>
        <i class="fa-solid fa-phone text-warning me-1"></i>
        <strong>${c.cedula}</strong> - ${c.nombre} →
        <span class="text-primary fw-bold">Llamar al ${c.telefono}</span>
      </li>
    `;
  }
});

html += `
  </tbody>
</table>
</div>
`;

contenedor.innerHTML = html;

```

```

const divRecordatorio =
document.getElementById("recordatorioLlamadas");
const ulRecordatorio = document.getElementById("listaRecordatorios");

if (tieneSinCorreo) {
    ulRecordatorio.innerHTML = recordatorios;
    divRecordatorio.classList.remove("d-none");
    setTimeout(() => {
        document.querySelectorAll(".btnRegistrarLlamada").forEach((btn) =>
{
    btn.addEventListener("click", async function () {
        const idCliente = this.dataset.id;
        const nombre = this.dataset.nombre;
        const telefono = this.dataset.telefono;

        const confirm = await Swal.fire({
            title: "¿Registrar llamada?",
            text: `¿Confirmás que llamaste a ${nombre} al número
${telefono}?`,
            icon: "question",
            showCancelButton: true,
            confirmButtonText: "Sí, registrar",
            cancelButtonText: "Cancelar",
        });

        if (confirm.isConfirmed) {
            const formData = new URLSearchParams();
            formData.append("action", "registrarLlamadaCumple");
            formData.append("idCliente", idCliente);

            try {
                const res = await fetch(
                    "/CRM_INT/CRM/controller/CumpleController.php",
                    {
                        method: "POST",
                        headers: {
                            "Content-Type": "application/x-www-form-urlencoded",
                        },
                        body: formData.toString(),
                    }
                );
                const data = await res.json();

                if (data.success) {
                    Swal.fire("¡Llamada registrada!", data.message,
"success");
                    cargarCumples();
                    cargarHistorial();
                }
            } catch (error) {
                console.error("Error al registrar la llamada:", error);
            }
        }
    });
}
});
```

```

        if (window.actualizarCumpleBadgeSidebar)
            window.actualizarCumpleBadgeSidebar();
        // Limpia el recordatorio por si acaso
        document.getElementById("listaRecordatorios").innerHTML =
        "";
        document
            .getElementById("recordatorioLlamadas")
            .classList.add("d-none");
    } else {
        Swal.fire("Error", data.message, "error");
    }
} catch (err) {
    console.error("Error al registrar llamada:", err);
    Swal.fire(
        "Error",
        "No se pudo conectar con el servidor",
        "error"
    );
}
}
});
});
},
300); // Espera un poco para asegurar que los botones ya existen
en el DOM
} else {
    divRecordatorio.classList.add("d-none");
    ulRecordatorio.innerHTML = "";
}
};

const seleccionarCumple = (id, nombre, cedula, correo, telefono, fecha)
=> {
    document.getElementById("idCumple").value = id;
    document.getElementById("nombreCorreo").value = nombre;
    document.getElementById("cedulaCorreo").value = cedula;
    document.getElementById("correoCorreo").value = correo;
    document.getElementById("telefonoCorreo").value = telefono;
    document.getElementById("fechaCumple").value = fecha;

    const btn = document.getElementById("btnEnviarCorreo");
    const btnWhats2 = document.getElementById("btnEnviarWhats");
    if (!correo) {
        Swal.fire({
            icon: "warning",
            title: "¡Este cliente no tiene correo!",
            text: "Recordá llamarlo o escribirle un mensaje.",
            confirmButtonText: "Entendido",
        });
    }
}

```

```

btn.disabled = true;
if (btnWhats2) btnWhats2.disabled = !telefono;
} else {
  btn.disabled = false;
  if (btnWhats2) btnWhats2.disabled = !telefono;

  // NUEVO: Focus inmediato al botón
  setTimeout(() => {
    // Scroll suave al botón
    btn.scrollIntoView({
      behavior: "smooth",
      block: "center",
      inline: "nearest",
    });
  });

  // Focus y efecto visual
  setTimeout(() => {
    btn.focus();
    btn.style.transition = "all 0.3s ease";
    btn.style.transform = "scale(1.02)";
    btn.style.boxShadow = "0 0 20px rgba(249, 196, 31, 0.7)";

    // Restaurar estilo después de 1.5 segundos
    setTimeout(() => {
      btn.style.transform = "scale(1)";
      btn.style.boxShadow = "";
    }, 1500);
    }, 600);
  }, 200);
}

const cambiarEstado = async (id, nuevoEstado) => {
  const formData = new URLSearchParams();
  formData.append("action", "cambiarEstado");
  formData.append("id", id);
  formData.append("estado", nuevoEstado);

  try {
    const res = await
    fetch("/CRM_INT/CRM/controller/CumpleController.php", {
      method: "POST",
      headers: { "Content-Type": "application/x-www-form-urlencoded" },
      body: formData.toString(),
    });
  }

  const data = await res.json();
  return data.success;
}

```

```

} catch (error) {
    console.error("Error en cambiarEstado:", error);
    return false;
}

};

const formatearFecha = (fechaStr) => {
    if (!fechaStr) return "Fecha no disponible";

    const partes = fechaStr.split("-");
    if (partes.length !== 3) return "Fecha inválida";

    const [anio, mes, dia] = partes;
    const fecha = new Date(anio, mes - 1, dia);

    if (isNaN(fecha)) return "Fecha inválida";

    return fecha
        .toLocaleDateString("es-CR", {
            day: "2-digit",
            month: "long",
            year: "numeric",
        })
        .toUpperCase();
};

function cargarHistorial() {
    fetch("/CRM_INT/CRM/controller/CumpleController.php", {
        method: "POST",
        headers: { "Content-Type": "application/x-www-form-urlencoded" },
        body: "action=readHistorial",
    })
    .then((res) => res.json())
    .then((data) => {
        let html = `
            <div class="table-responsive">
                <table class="table table-striped table-bordered"
id="tablaHistorial">
                    <thead class="table-dark">
                        <tr>
                            <th>Cédula</th>
                            <th>Nombre</th>
                            <th>CORREO / TELÉFONO</th>
                            <th>Fecha de Cumpleaños</th>
                            <th>Fecha de Llamada/Correo</th>
                            <th>Vence</th>
                            <th>Vencido</th>
                        </tr>
                    </thead>

```

```

        <tbody>
        `;

        data.forEach((c) => {
            let datoContacto =
                c.correo && c.correo.trim() !== "" ? c.correo : c.telefono;
            html += `
                <tr>
                    <td>${c.cedula}</td>
                    <td>${c.nombre}</td>
                    <td class="contacto">${datoContacto}</td>
                    <td>${c.fechaCumpleanos}</td>
                    <td>${c.fechaLlamada}</td>
                    <td>${c.vence}</td>
                    <td class="vencido-cell">${c.vencido === "SI" ? "SÍ" :
                    "NO"}</td>
                </tr>
            `;
        });

        html += `
            </tbody>
            </table>
        </div>
        `;

        document.getElementById("historialCumple").innerHTML = html;
    });
}

document.getElementById("btnImprimir").addEventListener("click", () => {
    const tabla = document.getElementById("tablaHistorial");

    if (!tabla || tabla.rows.length <= 1) {
        Swal.fire(
            "Tabla vacía",
            "No hay datos en el historial para imprimir.",
            "warning"
        );
        return;
    }

    const contenido =
        document.getElementById("historialCumple").innerHTML;
    const ventana = window.open("", "", "height=800,width=1000");

    ventana.document.write("<html><head><title>Historial de
Cumpleaños</title>");
}

```

```

ventana.document.write(`

<style>
  body {
    font-family: Arial, sans-serif;
    margin: 20px;
  }
  table {
    width: 100%;
    border-collapse: collapse;
    font-size: 13px;
  }
  th, td {
    border: 1px solid black;
    padding: 6px;
    text-align: center;
  }
  th {
    background-color: #f9c41f;
    color: black;
  }
</style>
`);

ventana.document.write("</head><body>");
ventana.document.write(
  "<h3 style='text-align:center;'>Historial de Cumpleaños
Atendidos</h3>";
);
ventana.document.write(contenido);
ventana.document.write("</body></html>");
ventana.document.close();

ventana.focus();
ventana.print();
ventana.close();
});

document.getElementById("btnExportPDF").addEventListener("click",
function () {
  const tabla = document.getElementById("tablaHistorial");

  if (!tabla || tabla.rows.length <= 1) {
    Swal.fire(
      "Tabla vacía",
      "No hay datos en el historial para exportar.",
      "warning"
    );
    return;
}

```

```

const { jsPDF } = window.jsPDF;
const doc = new jsPDF({
    orientation: "landscape",
    unit: "pt",
    format: "a4",
});

// Título
doc.setFont("helvetica", "bold");
doc.setFontSize(16);
doc.text(
    "Historial de Cumpleaños Atendidos",
    doc.internal.pageSize.getWidth() / 2,
    40,
    {
        align: "center",
    }
);

// Extraer la tabla directamente
doc.autoTable({
    html: "#tablaHistorial",
    startY: 60,
    styles: {
        font: "helvetica",
        fontSize: 10,
        cellPadding: 4,
    },
    headStyles: {
        fillColor: [249, 196, 31], // amarillo institucional
        textColor: [0, 0, 0],
        halign: "center",
    },
    bodyStyles: {
        halign: "center",
    },
    didDrawPage: function (data) {
        // Footer con fecha
        const fecha = new Date().toLocaleString("es-CR");
        doc.setFontSize(8);
        doc.text(
            `CRM Bastos - ${fecha}`,
            data.settings.margin.left,
            doc.internal.pageSize.getHeight() - 10
        );
    },
});

```

```

    doc.save("historial_cumpleanos.pdf");
});

// === MEJORAS PARA EL SISTEMA DE CUMPLEAÑOS ===

// 1. BUSCADOR POR CÉDULA PARA EL HISTORIAL
function agregarBuscadorHistorial() {
    // Buscar el contenedor donde vamos a agregar el input
    const historialContainer = document.getElementById("historialCumpleas");
    if (!historialContainer) return;

    // Verificar si ya existe el buscador para evitar duplicados
    if (document.getElementById("buscadorCedula")) return;

    // Crear el input de búsqueda
    const buscadorHTML = `
        <div class="mb-3">
            <div class="input-group input-group-sm">
                <input
                    type="text"
                    id="buscadorCedula"
                    class="form-control"
                    placeholder="Buscar por cédula (ej: 504590528)...""
                    autocomplete="off"
                >
                <button
                    type="button"
                    class="btn btn-outline-secondary"
                    id="limpiarBusqueda"
                    title="Limpiar búsqueda">
                    <i class="fas fa-times"></i>
                </button>
            </div>
            <small class="text-muted">
                <i class="fas fa-info-circle"></i>
                Escribí la cédula para filtrar los resultados
            </small>
        </div>
    `;

    // Insertar el buscador antes del contenido de la tabla
    historialContainer.insertAdjacentHTML("afterbegin", buscadorHTML);

    // Event listeners para el buscador
    const inputBuscador = document.getElementById("buscadorCedula");
    const btnLimpiar = document.getElementById("limpiarBusqueda");

```

```

// Función de filtrado
function filtrarTabla() {
    const termino = inputBuscador.value.toLowerCase().trim();
    const tabla = document.getElementById("tablaHistorial");

    if (!tabla) return;

    const filas = tabla
        .getElementsByTagName("tbody")[0]
        .getElementsByTagName("tr");
    let filasVisibles = 0;

    for (let i = 0; i < filas.length; i++) {
        const celdaCedula = filas[i].getElementsByTagName("td")[0]; // Primera columna (Cédula)

        if (celdaCedula) {
            const textoCedula = celdaCedula.textContent.toLowerCase();

            if (termino === "" || textoCedula.includes(termino)) {
                filas[i].style.display = "";
                filasVisibles++;
            } else {
                filas[i].style.display = "none";
            }
        }
    }

    // Mostrar mensaje si no hay resultados
    mostrarMensajeResultados(filasVisibles, termino);
}

// Función para mostrar mensaje de resultados
function mostrarMensajeResultados(cantidad, termino) {
    // Remover mensaje anterior si existe
    const mensajeAnterior = document.getElementById("mensajeResultados");
    if (mensajeAnterior) {
        mensajeAnterior.remove();
    }

    // Si hay término de búsqueda, mostrar contador
    if (termino !== "") {
        const mensaje = document.createElement("div");
        mensaje.id = "mensajeResultados";
        mensaje.className =
            cantidad > 0
            ? "alert alert-info alert-sm py-1 px-2 mb-2"
            : "alert alert-warning alert-sm py-1 px-2 mb-2";
    }
}

```

```

mensaje.innerHTML =
    cantidad > 0
        ? `<small><i class="fas fa-filter"></i> Se encontraron
<strong>${cantidad}</strong> resultados para
<strong>${termino}</strong>"</small>` :
        : `<small><i class="fas fa-exclamation-triangle"></i> No se
encontraron resultados para "<strong>${termino}</strong>"</small>`;
    }

    inputBuscador.parentNode.parentNode.insertAdjacentElement(
        "afterend",
        mensaje
    );
}

// Event listeners
inputBuscador.addEventListener("input", filtrarTabla);
inputBuscador.addEventListener("keyup", function (e) {
    if (e.key === "Escape") {
        this.value = "";
        filtrarTabla();
        this.blur();
    }
});
btnLimpiar.addEventListener("click", function () {
    inputBuscador.value = "";
    filtrarTabla();
    inputBuscador.focus();
});
}

// 2. FUNCIÓN PARA HACER SCROLL Y FOCUS AL FORMULARIO
function scrollYFocusFormulario() {
    // Primero hacer scroll al formulario
    const formularioCard = document.querySelector(
        '.card.shadow-sm[style*="border-left: 5px solid #f9c41f"]'
    );

    if (formularioCard) {
        // Scroll suave al formulario
        formularioCard.scrollIntoView({
            behavior: "smooth",
            block: "center",
            inline: "nearest",
        });
    }

    // Después del scroll, hacer focus al botón
}

```

```

setTimeout(() => {
    const btnEnviarCorreo = document.getElementById("btnEnviarCorreo");
    if (btnEnviarCorreo) {
        // Añadir efecto visual
        btnEnviarCorreo.style.transition = "all 0.3s ease";
        btnEnviarCorreo.style.transform = "scale(1.05)";
        btnEnviarCorreo.style.boxShadow = "0 0 20px rgba(249, 196, 31, 0.6)";

        // Focus al botón
        btnEnviarCorreo.focus();

        // Remover efecto después de 2 segundos
        setTimeout(() => {
            btnEnviarCorreo.style.transform = "scale(1)";
            btnEnviarCorreo.style.boxShadow =
                "0 4px 16px rgba(249, 196, 31, 0.3)";
        }, 2000);
    }
}, 800); // Esperar a que termine el scroll
}

// Modificar la función cargarHistorial para agregar el buscador
const cargarHistorialOriginal = cargarHistorial;
cargarHistorial = function () {
    cargarHistorialOriginal.call(this);

    // Agregar el buscador después de que se cargue la tabla
    setTimeout(() => {
        agregarBuscadorHistorial();
    }, 100);
};

// Modificar la función seleccionarCumple para agregar el scroll/focus
const seleccionarCumpleOriginal = seleccionarCumple;
seleccionarCumple = function (id, nombre, cedula, correo, telefono, fecha) {
    // Ejecutar la función original
    seleccionarCumpleOriginal.call(
        this,
        id,
        nombre,
        cedula,
        correo,
        telefono,
        fecha
    );
}

```

```

// Agregar el scroll y focus al formulario
setTimeout(() => {
    scrollYFocusFormulario();
}, 300); // Pequeña pausa para que se complete la selección
};

```

### |   |— Dashboard.js

```

$(document).ready(function () {
    // -----
    // ◇ Variables principales
    // -----
    const $inputDescripcion = $("#descripcion");           // Input de
    descripción de tarea
    const $contador = $("#contador-caracteres");          // Contador de
    caracteres

    const nombreUsuario = localStorage.getItem("nombreUsuario"); // Nombre
    guardado en localStorage
    const rolUsuario = localStorage.getItem("rolUsuario");      // Rol
    guardado en localStorage

    // -----
    // ◇ Mostrar nombre y rol en bienvenida
    // -----
    if (nombreUsuario) {
        $("#bienvenida-nombre").text(nombreUsuario);
    }
    if (rolUsuario) {
        $("#bienvenida-rol").text(rolUsuario);
    }

    // -----
    // ◇ Contador de caracteres (máx 220)
    // -----
    $inputDescripcion.on("input", function () {
        const max = 220;
        const restante = max - $(this).val().length; // Resta caracteres
        usados
        $contador.text(restante);                  // Actualiza contador
    });

    // -----
    // ◇ Obtener clientes vía AJAX
    // -----
    $.ajax({

```

```

url: "/CRM_INT/CRM/controller/ClienteController.php?action=readAll",
type: "GET",
dataType: "json",
success: function (res) {
    if (res.success && res.data) {
        const clientes = res.data;

        // ↵ Total de clientes
        $("#total-clientes").text(clientes.length.toLocaleString());

        // ↵ Cliente del mes (el que más acumulado tiene)
        const clienteMes = clientes
            .filter(
                (c) =>
                    c.acumulado && !isNaN(c.acumulado) &&
parseFloat(c.acumulado) > 0
            )
            .reduce(
                (max, cliente) =>
                    cliente.acumulado > max.acumulado ? cliente : max,
                { acumulado: 0 }
            );

        if (clienteMes && clienteMes.nombre) {
            $("#cliente-mes-nombre").text(clienteMes.nombre);
            $("#cliente-mes-
valor").text(formatearColones(clienteMes.acumulado));
        } else {
            $("#cliente-mes-nombre").text("N/A");
            $("#cliente-mes-valor").text(formatearColones(0));
        }

        // ↵ Clientes que cumplen años en el mes actual
        const mesActual = new Date().getMonth() + 1;
        const cumpleaneros = clientes.filter((cliente) => {
            if (!cliente.fechaCumpleanos) return false;
            const mes = new Date(cliente.fechaCumpleanos).getMonth() + 1;
            return mes === mesActual;
        });

        const limite = 3;
        if (cumpleaneros.length > 0) {
            const nombres = cumpleaneros.slice(0, limite).map((c) =>
c.nombre);
            const textoFinal =
                nombres.join(", ") +
                (cumpleaneros.length > limite ? " y más..." : "");
            $("#cumple-texto")
        }
    }
}

```

```

        .text(textoFinal)
        .attr("title", cumpleaneros.map((c) => c.nombre).join(", "));
    } else {
        $("#cumple-texto").text("No hay cumpleaños este mes");
    }
},
error: function (xhr, status, error) {
    console.error("Error al obtener clientes:");
    console.log("Status:", status);
    console.log("Error:", error);
    console.log("Respuesta del servidor:", xhr.responseText);
},
});

// -----
// ⚡ Obtener datos de ventas vía AJAX
// -----
$.ajax({
    url: "/CRM_INT/CRM/controller/CompraController.php",
    method: "POST",
    data: { action: "readAll" },
    dataType: "json",
    success: function (response) {
        if (response.success && response.data) {
            const compras = response.data;
            const fechaActual = new Date();
            const mesActual = fechaActual.getMonth() + 1;
            const anioActual = fechaActual.getFullYear();

            // ⚡ Filtrar compras del mes actual
            const comprasMesActual = compras.filter((compra) => {
                if (!compra.fechaCompra) return false;
                const fechaCompra = new Date(compra.fechaCompra);
                return (
                    fechaCompra.getFullYear() === anioActual &&
                    fechaCompra.getMonth() === mesActual - 1
                );
            });

            // ⚡ Filtrar compras del año actual
            const comprasAnioActual = compras.filter((compra) => {
                if (!compra.fechaCompra) return false;
                const fechaCompra = new Date(compra.fechaCompra);
                return fechaCompra.getFullYear() === anioActual;
            });

            // ⚡ Totales
        }
    }
});

```

```

        const totalVentasMes = comprasMesActual.reduce(
            (total, compra) => total + parseFloat(compra.total || 0),
            0
        );

        const totalVentasAnio = comprasAnioActual.reduce(
            (total, compra) => total + parseFloat(compra.total || 0),
            0
        );

        // ⚡ Actualizar en el DOM
        $("#total-ventas").text(formatearColones(totalVentasMes));
        $("#total-ventas-anio").text(formatearColones(totalVentasAnio));

        console.log(
            `Ventas del mes ${mesActual}/${anioActual}:`,
            formatearColones(totalVentasMes)
        );
        console.log(
            `Ventas del año ${anioActual}:`,
            formatearColones(totalVentasAnio)
        );
    },
    error: function (xhr, status, error) {
        console.error("Error al obtener datos de ventas:");
        console.log("Status:", status);
        console.log("Error:", error);
        console.log("Respuesta del servidor:", xhr.responseText);
    },
});

// -----
// ◇ Resetear contador de caracteres
// -----
function resetearContador() {
    $contador.text(220);
}

// -----
// ◇ Resetear contador al enviar tarea
// -----
$(document).on("submit", "#formTarea", function () {
    setTimeout(function () {
        if ($inputDescripcion.val() === "") {
            resetearContador(); // Solo si el input se limpió
        }
    }, 100);
}

```

```

    });
});

// -----
// ◇ Función auxiliar: formatear a colones (₡)
// -----
function formatearColones(valor) {
    return new Intl.NumberFormat("es-CR", {
        style: "currency",
        currency: "CRC",
        minimumFractionDigits: 0,
    }).format(valor);
}

```

|   |— jquery-3.6.3.min.js

Descargarlo en cualquier navegador, es demasiado extenso

|   |— Layout.js

```

// Archivo: public/js/Layout.js
// Detectar cuando se cierra la ventana/pestana y cerrar sesión

// Flag para saber si es navegación interna (dentro del CRM)
window.__navInterna = false;

document.addEventListener("DOMContentLoaded", function () {
    // ◇ Función para enviar cierre de sesión al servidor
    const sendClose = () =>
        navigator.sendBeacon(
            "/CRM_INT/CRM/controller/SessionController.php",
            JSON.stringify({ action: "close_session" })
        );

    // ◇ Evento al intentar salir/cerrar pestana
    window.addEventListener("beforeunload", function () {
        if (window.__navInterna) return;
        sendClose();
        if ("scrollRestoration" in history) history.scrollRestoration =
            "manual";
    });

    // ◇ Evento al descargar página (unload)
    window.addEventListener("unload", function () {
        if (window.__navInterna) return;
        sendClose();
    });

    // ◇ Mover modales al body para evitar problemas visuales
    (function ensureModalsAtBody() {

```

```

[ "modalTarjeta", "modalCumple" ].forEach((id) => {
    const el = document.getElementById(id);
    if (el && el.parentElement !== document.body) {
        document.body.appendChild(el);
    }
});
})();
});

// -----
// ◇ Confirmar cierre manual de sesión
// -----

function confirmarCerrarSesion() {
    if (confirm("¿Cerrar sesión?")) {
        window.location.href = "/CRM_INT/CRM/index.php?logout=1";
    }
}

window.addEventListener("load", function () {
    const toggleBtn = document.getElementById("menuToggle"); // Botón ☰
    const sidebar = document.querySelector(".sidebar"); // Sidebar
    const overlay = document.getElementById("sidebarOverlay"); // Fondo oscuro
    const sidebarLinks = document.querySelectorAll(".sidebar ul li a");

    // ⚠️ El scroll real está en el <ul> del sidebar
    const sc = sidebar ? sidebar.querySelector("ul") || sidebar : null;

    if (!toggleBtn || !sidebar) {
        console.warn("⚠️ No se encontró el botón ☰ o el sidebar.");
        return;
    }

    // ◇ Restaurar posición de scroll guardada
    const savedScrollPosition =
localStorage.getItem("sidebarScrollPosition");
    if (savedScrollPosition && sc) {
        sc.scrollTop = parseInt(savedScrollPosition, 10);
    }

    if (sc) {
        sc.addEventListener("scroll", function () {
            localStorage.setItem("sidebarScrollPosition", sc.scrollTop);
        });
    }

    // ◇ Centrar en vista el elemento activo del sidebar
    function centrarElementoActivo() {

```

```

const elementoActivo = sidebar.querySelector(".active");
if (!elementoActivo || !sc) return;

const elRect = elementoActivo.getBoundingClientRect();
const contRect = sc.getBoundingClientRect();
const elTopWithin = elRect.top - contRect.top + sc.scrollTop;

const target =
  elTopWithin - sc.clientHeight / 2 + elementoActivo.clientHeight /
2;

sc.scrollTo({
  top: Math.max(0, target),
  behavior: "instant" in HTMLElement.prototype ? "instant" : "auto",
});

localStorage.setItem("sidebarScrollPosition", sc.scrollTop);
}

// ◇ Restaurar Último foco del sidebar al cargar
setTimeout(() => {
  const lastId = localStorage.getItem("lastSidebarFocusId");
  const activo = sidebar.querySelector(".active");

  const objetivo = (lastId && document.getElementById(lastId)) ||
activo;
  if (!objetivo) return;

  const prev = sc.scrollTop;
  try {
    objetivo.focus({ preventScroll: true });
  } catch {
    objetivo.focus();
  }
  if (isGuia) sc.scrollTop = prev; // solo en “guia” forzamos posición
}, 180);

// ◇ Forzar sidebar cerrado en móvil
if (window.innerWidth <= 992) {
  sidebar.classList.remove("show");
  sidebar.classList.remove("activa");
  if (overlay) overlay.classList.remove("show");
}

// ◇ Toggle del menú lateral
toggleBtn.addEventListener("click", function () {
  sidebar.classList.toggle("show");
  sidebar.classList.toggle("activa");
}

```

```

        if (overlay) {
            overlay.classList.toggle("show");
        }
    });

// ◇ Cerrar al hacer click en overlay
if (overlay) {
    overlay.addEventListener("click", function () {
        sidebar.classList.remove("show");
        sidebar.classList.remove("activa");
        overlay.classList.remove("show");
    });
}

// ◇ Cerrar sidebar al hacer click en un link
sidebarLinks.forEach((link) => {
    link.addEventListener("click", function () {
        window.__navInterna = true;
        if (this.id) localStorage.setItem("lastSidebarFocusId", this.id);
        if (sc) localStorage.setItem("sidebarScrollPosition",
sc.scrollTop);

        if (window.innerWidth <= 992) {
            sidebar.classList.remove("show");
            sidebar.classList.remove("activa");
            if (overlay) overlay.classList.remove("show");
        }
    });
});

// ◇ Cerrar al redimensionar ventana
window.addEventListener("resize", function () {
    if (window.innerWidth > 992) {
        sidebar.classList.remove("show");
        sidebar.classList.remove("activa");
        if (overlay) overlay.classList.remove("show");
    }
});
});

// -----
// ◇ Modal TARJETA (buscar por número)
// -----
function mostrarModal() {
    const modal = document.getElementById("modalTarjeta");
    if (!modal) return;
    document.body.classList.add("modal-open"); // bloquea scroll
    modal.style.display = "flex";
}

```

```

const input = document.getElementById("modalInputTarjeta");
if (input) {
    input.value = "";
    const err = document.getElementById("modalMensajeError");
    if (err) err.textContent = "";
    setTimeout(() => input.focus(), 100);
}
}

function cerrarModal() {
    const modal = document.getElementById("modalTarjeta");
    if (!modal) return;
    modal.style.display = "none";
    const input = document.getElementById("modalInputTarjeta");
    if (input) input.value = "";
    const err = document.getElementById("modalMensajeError");
    if (err) err.textContent = "";

    // Quita modal-open si no queda otro modal abierto
    const cumpleAbierto = document.querySelector(
        "#modalCumple[style*='display: flex']"
    );
    if (!cumpleAbierto) document.body.classList.remove("modal-open");
}

// ◇ Alias por compatibilidad con HTML
function abrirModal() {
    mostrarModal();
}

// ◇ Buscar tarjeta y redirigir a compras
async function redirigirCompra() {
    const tarjeta = (document.getElementById("modalInputTarjeta")?.value ||
    "")?.trim();
    const mensajeError = document.getElementById("modalMensajeError");
    if (mensajeError) mensajeError.textContent = "";

    if (!tarjeta) {
        if (mensajeError)
            mensajeError.textContent = "Por favor ingrese un número de
tarjeta.";
        return;
    }

    try {
        const response = await
fetch("/CRM_INT/CRM/controller/ClienteController.php", {

```

```

        method: "POST",
        headers: { "Content-Type": "application/x-www-form-urlencoded" },
        body: `action=read&id=${encodeURIComponent(tarjeta)}`,
    });
const json = await response.json();

if (!json.success || !json.data) {
    if (mensajeError) mensajeError.textContent = "El número de tarjeta no existe.";
    return;
}

// Guarda foco antes de redirigir
localStorage.setItem("lastSidebarFocusId", "link-compras");

// Redirigir a vista de compras
window.location.href =
`/CRM_INT/CRM/index.php?view=compras&idCliente=${encodeURIComponent(
    tarjeta
)}&buscar=auto`;
} catch (e) {
    if (mensajeError) mensajeError.textContent = "Error de conexión con el servidor.";
}
}

// ◇ Eventos de cierre por click fuera y teclado
document.addEventListener("DOMContentLoaded", function () {
    const modal = document.getElementById("modalTarjeta");
    const inputTarjeta = document.getElementById("modalInputTarjeta");

    if (modal) {
        modal.addEventListener("click", function (e) {
            if (e.target === modal) cerrarModal();
        });

        document.addEventListener("keydown", function (e) {
            if (modal.style.display === "flex") {
                if (e.key === "Escape") cerrarModal();
                if (e.key === "Enter") {
                    e.preventDefault();
                    redirigirCompra();
                }
            }
        });
    }
}

if (inputTarjeta) {

```

```

        inputTarjeta.addEventListener("keydown", function (e) {
            if (e.key === "Enter") {
                e.preventDefault();
                redirigirCompra();
            }
        });
    });

// -----
// ◇ Modal CUMPLEAÑOS
// -----
function abrirModalCumple() {
    const modal = document.getElementById("modalCumple");
    if (modal) {
        document.body.classList.add("modal-open");
        modal.style.display = "flex";
        cargarCumpleanosSemana(); // Cargar datos al abrir
    } else {
        console.error("Modal modalCumple no encontrado en el DOM");
    }
}

function cerrarModalCumple() {
    const modal = document.getElementById("modalCumple");
    if (modal) {
        modal.style.display = "none";
        const tarjetaAbierto = document.querySelector(
            "#modalTarjeta[style*='display: flex']"
        );
        if (!tarjetaAbierto) document.body.classList.remove("modal-open");
    }
}

function irACumpleanos() {
    cerrarModalCumple();
    window.location.href = "/CRM_INT/CRM/index.php?view=cumple";
}

// ◇ Cargar cumpleaños de la semana
async function cargarCumpleanosSemana() {
    const contenedor = document.getElementById("listaCumpleanos");
    const rangoDiv = document.getElementById("rangoCumpleSemana");

    if (!contenedor || !rangoDiv) {
        console.error("No se encontraron los elementos del modal");
        return;
    }
}

```

```

mostrarRangoSemana(rangoDiv);

contenedor.innerHTML = `
  <div class="text-center p-3">
    <div class="spinner-border text-warning" role="status"></div>
    <p class="mt-2">Cargando cumpleaños...</p>
  </div>
`;

try {
  const response = await
fetch("/CRM_INT/CRM/controller/CumpleController.php", {
  method: "POST",
  headers: { "Content-Type": "application/x-www-form-urlencoded" },
  body: "action=readSemana",
});
}

const data = await response.json();

if (data.success) {
  renderizarCumpleanos(data.data, contenedor);
} else {
  contenedor.innerHTML = `<div class="alert alert-danger text-
center">${data.message}</div>`;
}
} catch (error) {
  console.error("Error cargando cumpleaños:", error);
  contenedor.innerHTML = `<div class="alert alert-danger text-
center">Error al cargar los cumpleaños.</div>`;
}

// ◇ Mostrar rango de la semana actual
function mostrarRangoSemana(div) {
  const hoy = new Date();
  const diaActual = hoy.getDay();
  const diffInicio = hoy.getDate() - diaActual + (diaActual === 0 ? -6 :
1);
  const lunes = new Date(hoy.setDate(diffInicio));
  const domingo = new Date(lunes);
  domingo.setDate(lunes.getDate() + 6);

  const opciones = { day: "2-digit", month: "long" };
  const formatoLunes = lunes.toLocaleDateString("es-CR", opciones);
  const formatoDomingo = domingo.toLocaleDateString("es-CR", opciones);
}

```

```

    div.innerHTML = `📅 Semana actual: <strong>${formatoLunes}</strong> al
<strong>${formatoDomingo}</strong>`;
}

// ◊ Renderizar lista de cumpleañeros
function renderizarCumpleanos(cumpleas, contenedor) {
    const pendientes = cumpleas.filter((c) => c.estado === "Activo");

    if (pendientes.length === 0) {
        contenedor.innerHTML =
            `

#### ¡No hay cumpleaños esta semana!



Todos los cumpleañeros han sido atendidos o no hay cumpleaños
                    programados.

`;
        return;
    }

    let html = "";

    pendientes.forEach((cumple) => {
        const tieneCorreo = cumple.correo && cumple.correo.trim() !== "";
        const cardClass = tieneCorreo ? "cumple-card" : "cumple-card sin-
correo";

        html += `
            <div class="${cardClass}">
                <div class="cumple-header">
                    <h4 class="cumple-nombre">${cumple.nombre}</h4>
                    <span class="cumple-fecha">${formatearFechaCumple(
                        cumple.fechaCumpleanos
                    )}</span>
                </div>

                <div class="cumple-info">
                    <div class="cumple-detalle">
                        <span class="cumple-label">Cédula:</span> ${cumple.cedula}
                    </div>
                    <div class="cumple-detalle">
                        <span class="cumple-label">Teléfono:</span>
                        ${cumple.telefono}
                    </div>
                    <div class="cumple-detalle">
                        <span class="cumple-label">Correo:</span> ${
                            tieneCorreo
                            ? cumple.correo
                        }
                    </div>
                </div>
            </div>
        `;
    });
}

```

```

        : '<span style="color: #ff6b6b;">Sin correo</span>'
    }
  </div>
</div>
</div>
`;
});

contenedor.innerHTML = html;
}

// ◇ Formatear fecha de cumpleaños
function formatearFechaCumple(fecha) {
  return new Date(fecha).toLocaleDateString("es-CR", {
    day: "2-digit",
    month: "short",
  });
}

// -----
// ◇ Badge de cumpleaños pendientes en sidebar
// -----
window.actualizarCumpleBadgeSidebar = function () {
  fetch("/CRM_INT/CRM/controller/CumpleController.php", {
    method: "POST",
    headers: { "Content-Type": "application/x-www-form-urlencoded" },
    body: "action=readSemana",
  })
  .then((res) => res.json())
  .then((data) => {
    const badge = document.getElementById("cumple-badge");
    if (!badge) return;
    if (data.success && data.data && data.data.length > 0) {
      badge.style.display = "inline-block";
      badge.innerHTML = `<span style="display:inline-block;width:12px;height:12px;background:#f9c41f;border-radius:50%;border:2px solid #000;box-shadow:0 0 2px #000;vertical-align:middle;"></span>`;
    } else {
      badge.style.display = "none";
      badge.innerHTML = "";
    }
  })
  .catch(() => {
    const badge = document.getElementById("cumple-badge");
    if (badge) {
      badge.style.display = "none";
      badge.innerHTML = "";
    }
  })
}

```

```

        }
    });

// -----
// ◇ Inicializar badge y eventos de modal cumple
// -----
document.addEventListener("DOMContentLoaded", function () {
    window.actualizarCumpleBadgeSidebar();

    const modalCumple = document.getElementById("modalCumple");
    if (modalCumple) {
        modalCumple.addEventListener("click", function (e) {
            if (e.target === modalCumple) {
                cerrarModalCumple();
            }
        });
    }

    document.addEventListener("keydown", function (e) {
        if (e.key === "Escape" && modalCumple.style.display === "flex") {
            cerrarModalCumple();
        }
    });
}
));

```

## |   |— Login.js

```

document.addEventListener("DOMContentLoaded", function () {
// -----
// ◇ Referencias a elementos del DOM
// -----
const loginForm = document.getElementById("loginForm");      // Formulario de login
const emailInput = document.getElementById("email");          // Campo usuario/email
const passwordInput = document.getElementById("password");   // Campo contraseña
const loginBtn = document.getElementById("loginBtn");         // Botón de login
const emailError = document.getElementById("emailError");    // Mensaje error usuario
const passwordError = document.getElementById("passwordError"); // Mensaje error contraseña

// -----
// ◇ Validaciones en inputs (usuario y contraseña)
// -----

```

```

emailInput.addEventListener("input", validateEmail);
emailInput.addEventListener("blur", validateEmail);
passwordInput.addEventListener("input", validatePassword);
passwordInput.addEventListener("blur", validatePassword);

// -----
// ◇ Evento submit del formulario
// -----
loginForm.addEventListener("submit", function (e) {
    e.preventDefault(); // Prevenir recarga

    if (loginBtn.disabled) return; // Evitar si el botón está
deshabilitado

    const isEmailValid = validateEmail();
    const isPasswordValid = validatePassword();

    if (isEmailValid && isPasswordValid) {
        handleLogin(); // Ejecuta login
    }
});

// -----
// ◇ Validación usuario
// -----
function validateEmail() {
    const email = emailInput.value.trim();
    const formGroup = emailInput.closest(".form-group");

    if (!email) {
        showError(emailError, "El usuario es requerido");
        setFormGroupState(formGroup, "error");
        return false;
    } else {
        hideError(emailError);
        setFormGroupState(formGroup, "success");
        return true;
    }
}

// -----
// ◇ Validación contraseña
// -----
function validatePassword() {
    const password = passwordInput.value;
    const formGroup = passwordInput.closest(".form-group");

    if (!password) {

```

```

        showError(passwordError, "La contraseña es requerida");
        setFormGroupState(formGroup, "error");
        return false;
    } else {
        hideError(passwordError);
        setFormGroupState(formGroup, "success");
        return true;
    }
}

// -----
// ◇ Mostrar/Ocultar errores
// -----
function showError(errorElement, message) {
    errorElement.textContent = message;
    errorElement.classList.add("show");
}

function hideError(errorElement) {
    errorElement.textContent = "";
    errorElement.classList.remove("show");
}

// -----
// ◇ Cambiar estado visual de form-group
// -----
function setFormGroupState(formGroup, state) {
    formGroup.classList.remove("success", "error");
    if (state) {
        formGroup.classList.add(state);
    }
}

// -----
// ◇ Lógica de login con fetch
// -----
function handleLogin() {
    // Limpiar rol antes de intentar login
    localStorage.removeItem("rolUsuario");
    const usuario = emailInput.value.trim();
    const contrasena = passwordInput.value;

    setLoadingState(true);          // Poner botón en estado cargando
    removeExistingErrors();         // Limpiar errores previos

    // Construir datos para enviar
    const formData = new FormData();
    formData.append("action", "login");
}

```

```

formData.append("usuario", usuario);
formData.append("contrasena", contrasena);

// Petición al backend
fetch("controller/UsuarioController.php", {
  method: "POST",
  body: formData,
  credentials: "same-origin",
})
.then((response) => response.text())
.then((text) => {
  console.log("📋 Texto recibido:", text);

  try {
    const jsonText = text.trim().match(/^\{.*\}$/s)?.[0]; // Buscar
JSON
    if (!jsonText) throw new Error("No se encontró JSON válido");

    const data = JSON.parse(jsonText);

    if (data.success) {
      // Guardar rol si existe
      if (data.rol) {
        localStorage.setItem("rolUsuario", data.rol);
      } else {
        localStorage.removeItem("rolUsuario");
      }

      // Guardar nombre si existe
      if (data.usuario) {
        localStorage.setItem("nombreUsuario", data.usuario);
      } else {
        localStorage.removeItem("nombreUsuario");
      }
    }

    showSuccessMessage(); // Mostrar mensaje de éxito

    // Redirigir después de 1.5s
    setTimeout(() => {
      window.location.href =
        data.redirect || "/CRM_INT/CRM/index.php?view=dashboard";
    }, 1500);
  } else {
    showLoginError(data.message || "Credenciales incorrectas");
  }
} catch (error) {
  console.error("✖ Error al parsear JSON:", error);
  showLoginError("Respuesta inválida del servidor:\n\n" + text);
}

```

```

        }
    })
    .catch((error) => {
        console.error("✖ Error de conexión:", error);
        showLoginError("Error de conexión. Por favor, intenta
nuevamente.");
    })
    .finally(() => {
        setLoadingState(false); // Volver a estado normal
    });
}

// -----
// ◇ Estado de carga en botón login
// -----
function setLoadingState(isLoading) {
    loginBtn.disabled = isLoading;
    loginBtn.classList.toggle("loading", isLoading);

    const btnText = loginBtn.querySelector(".btn-text");
    if (btnText) {
        btnText.textContent = isLoading ? "Verificando..." : "Iniciar
Sesión";
    }
}

// -----
// ◇ Mensaje éxito al loguear
// -----
function showSuccessMessage() {
    const btnText = loginBtn.querySelector(".btn-text");
    if (btnText) {
        const originalText = btnText.textContent;
        btnText.textContent = "¡Bienvenido!";
        loginBtn.style.background = "#28a745";
        loginBtn.style.color = "#fffff";

        setTimeout(() => {
            btnText.textContent = originalText;
            loginBtn.style.background = "";
            loginBtn.style.color = "";
        }, 3000);
    }
}

// -----
// ◇ Mostrar error de login
// -----

```

```

function showLoginError(message) {
    removeExistingErrors();

    const errorDiv = document.createElement("div");
    errorDiv.className = "login-error";
    errorDiv.textContent = message;
    errorDiv.style.cssText =
        `background: #f8d7da;
        color: #721c24;
        padding: 12px;
        border-radius: 4px;
        margin-bottom: 20px;
        border: 1px solid #f5c6cb;
        font-size: 14px;
        text-align: center;
        animation: fadeInUp 0.3s ease-out;
        `;

    loginForm.insertBefore(errorDiv, loginForm.firstChild);

    // Quitar mensaje después de 5s
    setTimeout(() => {
        if (errorDiv.parentNode) {
            errorDiv.style.animation = "fadeOut 0.3s ease-out";
            setTimeout(() => {
                if (errorDiv.parentNode) {
                    errorDiv.remove();
                }
            }, 300);
        }
    }, 5000);
}

// -----
// ◊ Eliminar errores previos
// -----
function removeExistingErrors() {
    const existingErrors = document.querySelectorAll(".login-error");
    existingErrors.forEach((error) => error.remove());
}

// -----
// ◊ Limpiar formulario (reset)
// -----
function clearForm() {
    emailInput.value = "";
    passwordInput.value = "";
    hideError(emailError);
}

```

```

        hideError(passwordError);
        setFormGroupState(emailInput.closest(".form-group"), "");
        setFormGroupState(passwordInput.closest(".form-group"), "");
        removeExistingErrors();
    }

    // Exponer función para usar globalmente
    window.clearLoginForm = clearForm;

    // -----
    // ◇ Atajos de teclado
    // -----
    document.addEventListener("keydown", function (e) {
        // Enter en email o password → enviar formulario
        if (
            e.key === "Enter" &&
            (e.target === emailInput || e.target === passwordInput)
        ) {
            e.preventDefault();
            loginForm.dispatchEvent(new Event("submit"));
        }

        // Escape → limpiar formulario
        if (e.key === "Escape") {
            clearForm();
        }
    });
}

// -----
// ◇ Animaciones y estilos dinámicos
// -----
const style = document.createElement("style");
style.textContent =
    `@keyframes fadeOut {
        from { opacity: 1; transform: translateY(0); }
        to { opacity: 0; transform: translateY(-10px); }
    }

    @keyframes fadeInUp {
        from { opacity: 0; transform: translateY(10px); }
        to { opacity: 1; transform: translateY(0); }
    }

    .form-group.focused label {
        color: var(--amarillo);
        transform: translateY(-2px);
        transition: all 0.3s ease;
    }
`;

```

```

.btn.loading {
    opacity: 0.7;
    cursor: not-allowed;
}

.btn.loading:hover {
    transform: none;
}

.login-error {
    border-radius: 8px;
    box-shadow: 0 2px 8px rgba(0,0,0,0.1);
}
`;  

document.head.appendChild(style);
});

```

## |   |— Manual.js

```

document.addEventListener("DOMContentLoaded", function () {
    // Efecto de escala al pasar el mouse
    document.querySelectorAll(".guia-card").forEach((card) => {
        card.addEventListener("mouseenter", () => {
            card.style.transform = "scale(1.01)";
        });
        card.addEventListener("mouseleave", () => {
            card.style.transform = "scale(1)";
        });
    });

    // Redirección desde los títulos
    document.querySelectorAll(".guia-link-seccion").forEach((enlace) => {
        enlace.style.cursor = "pointer";
        enlace.addEventListener("click", function () {
            const destino = this.getAttribute("data-ir-a");

            if (destino) {
                if (destino.startsWith("#")) {
                    const target = document.querySelector(destino);
                    if (target) {
                        target.scrollIntoView({ behavior: "smooth" });
                    }
                } else {
                    window.location.href = destino;
                }
            }
        });
    });
}

```

```

        });
    });

    // Lazy loading para videos de YouTube
    document.querySelectorAll(".video-placeholder").forEach((placeholder)
=> {
    placeholder.addEventListener("click", function() {
        const videoId = this.getAttribute("data-video-id");
        const iframe = document.createElement("iframe");

        iframe.setAttribute("width", "100%");
        iframe.setAttribute("height", "200");
        iframe.setAttribute("src",
`https://www.youtube.com/embed/${videoId}?autoplay=1`);
        iframe.setAttribute("frameborder", "0");
        iframe.setAttribute("allowfullscreen", "");
        iframe.setAttribute("allow", "autoplay");
        iframe.style.borderRadius = "8px";

        // Reemplazar el placeholder con el iframe
        this.parentNode.replaceChild(iframe, this);
    });
});
});

```

## |   └─ ScannerGlobal.js

```

// Scanner Global para códigos de barras - Funciona en todo el sistema
(() => {
    const REDIRECT = '/CRM_INT/CRM/index.php?view=compras&idCliente=';
    let buffer = '';
    let lastKeyTime = Date.now();
    let isScan = false;
    let timer;

    function reset() {
        buffer = '';
        isScan = false;
        clearTimeout(timer);
    }

    function procesar(code) {
        if (!code) return;
        const limpio = parseInt(code, 10).toString(); // quita ceros
        a la izquierda
        window.location.href =
`${REDIRECT}${encodeURIComponent(limpio)}&buscar=auto`;
    }
}

```

```

document.addEventListener('keydown', e => {
    // DEBUG: Mostrar información sobre el contexto actual
    const currentView = new
URLSearchParams(window.location.search).get('view');
    const currentPath = window.location.pathname;

    // NO PROCESAR si el usuario está escribiendo en un input, textarea,
o elemento editable
    const activeElement = document.activeElement;
    if (activeElement && (
        activeElement.tagName === 'INPUT' ||
        activeElement.tagName === 'TEXTAREA' ||
        activeElement.tagName === 'SELECT' ||
        activeElement.isContentEditable ||
        activeElement.closest('[contenteditable="true"]') ||
        activeElement.closest('.modal') // No procesar en modales
    )) {
        console.log(` Scanner deshabilitado - usuario escribiendo en:
${activeElement.tagName}`);
        return; // Salir sin procesar
    }

    // DEBUG: Confirmar que el scanner está activo
    console.log(` Scanner ACTIVO en: ${currentView || 'dashboard'} -
Tecla: ${e.key}`);

    const now = Date.now();
    const diff = now - lastKeyTime;
    lastKeyTime = now;

    // LÓGICA MEJORADA: Distinguir entre scanner y tecleo manual
    // - Scanner: teclas muy rápidas (<30ms) y consistentes
    // - Manual: más lento y errático
    if (diff < 30) {
        isScan = true; // Definitivamente es scanner
    } else if (diff > 300) {
        reset(); // Pausa muy larga = tecleo manual o nueva secuencia
        return;
    }

    if (e.key === 'Enter') {
        e.preventDefault();
        // SOLO procesar si realmente parece código de scanner
        if (buffer && isScan && buffer.length >= 8) { // Códigos de barras
suelen ser 8+ dígitos
            console.log(` Procesando código de scanner: ${buffer}`);
            procesar(buffer);
        } else {
    
```

```

        console.log(` Ignorado - no parece código válido: "${buffer}"`)

(len=${buffer.length}, isScan=${isScan}));
    }
    reset();
    return;
}

if (e.key.length === 1 && /[0-9]/.test(e.key)) { // SOLO números para
códigos de barras
    buffer += e.key;
    clearTimeout(timer);
    timer = setTimeout(() => {
        // Solo procesar si parece realmente un código de scanner
        if (buffer.length >= 8 && isScan) {
            console.log(`Timeout - procesando código: ${buffer}`);
            procesar(buffer);
        } else {
            console.log(`Timeout - ignorado: ${buffer}`);
(len=${buffer.length}, isScan=${isScan})`);
        }
        reset();
    }, 150);
} else if (e.key.length === 1) {
    // Si hay letras mezcladas, probablemente es tecleo manual
    console.log(` Letra detectada "${e.key}" - probablemente tecleo
manual`);
    reset();
}
});

// Limpia si se pierde el foco de la pestaña
window.addEventListener('blur', reset);

})();

```

## |   |— Tarea.js

```

$(document).ready(function () {
// -----
// ◇ Variable global de tareas
// -----
let tareas = [];

// -----
// ◇ Listar todas las tareas desde el servidor
// -----
function listarTareas() {
$.ajax({

```

```

url: "/CRM_INT/CRM/controller/TareaController.php?action=readAll",
type: "GET",
dataType: "json",
success: function (response) {
    if (response.success && response.data) {
        tareas = response.data; // Guardar tareas en memoria
        renderizarTareas(); // Dibujar tareas en pantalla
    } else {
        console.warn("No se encontraron tareas.");
    }
},
error: function (xhr, status, error) {
    console.error(" Error al cargar tareas:", error);
},
});
}

// -----
// ◊ Renderizar las tareas en la lista del DOM
// -----
function renderizarTareas() {
    const ul = $("#contenedorTarjetas");
    ul.empty(); // Limpiar lista antes de redibujar

    if (tareas.length === 0) {
        ul.append(`<li class="todo-list-item">No hay tareas
registradas.</li>`);
        return;
    }

    tareas.forEach((t) => {
        // Estado de la tarea (pendiente o completada)
        const estadoClase = t.estado === "completada" ? "estado-completada"
: "estado-pendiente";
        const estadoTexto = t.estado === "completada" ? "Completada" :
"Pendiente";

        // Formatear fecha si existe
        let fechaStr = "";
        if (t.fechaCreacion) {
            const fecha = new Date(t.fechaCreacion);
            fechaStr = fecha.toLocaleDateString("es-CR", {
                day: "2-digit",
                month: "2-digit",
                year: "numeric",
            });
        }
    })
}

```

```

// Plantilla HTML de cada tarea
const item = `
    <li class="todo-list-item">
        <div class="contenido-tarea">
            <div class="descripcion-tarea">${t.descripcion}</div>
            <div class="info-tarea" style="margin-top: 0.5rem; display:
flex; align-items: center; gap: 10px;">
                <span class="${estadoClase}">${estadoTexto}</span>
                ${fechaStr ? `<span class="fecha-tarea">${fechaStr}</span>` :
' '}
            </div>
        </div>
        <div>
            <button class="btn-cambiar" data-id="${t.id}" data-
estado="${t.estado}">Cambiar estado</button>
            <button class="btn-eliminar" data-
id="${t.id}">Eliminar</button>
        </div>
    </li>
`;
ul.append(item);
});

}

// -----
// ◇ Crear nueva tarea (formulario)
// -----
$("#formTarea").on("submit", function (e) {
    e.preventDefault();
    const descripcion = $("#descripcion").val().trim();
    if (!descripcion) return; // Evita tareas vacías

    $.ajax({
        url: "/CRM_INT/CRM/controller/TareaController.php?action=create",
        type: "POST",
        data: { descripcion },
        dataType: "json",
        success: function (res) {
            if (res.success) {
                $("#descripcion").val(""); // Limpiar input
                listarTareas(); // Recargar lista
            } else {
                alert("No se pudo agregar la tarea.");
            }
        },
        error: function () {
            console.error(" Error al insertar tarea.");
        },
    });
}

```

```

    });
});

// -----
// ◇ Cambiar estado de una tarea
// -----
$("#contenedorTarjetas").on("click", ".btn-cambiar", function () {
    const id = $(this).data("id");
    const estado = $(this).data("estado");
    const nuevoEstado = estado === "pendiente" ? "completada" :
"pendiente";

    $.ajax({
        url: "/CRM_INT/CRM/controller/TareaController.php?action=update",
        type: "POST",
        data: { id, estado: nuevoEstado },
        dataType: "json",
        success: function (res) {
            if (res.success) listarTareas();
            else alert("Error al cambiar el estado.");
        },
    });
});

// -----
// ◇ Eliminar una tarea
// -----
$("#contenedorTarjetas").on("click", ".btn-eliminar", function () {
    const id = $(this).data("id");
    if (!confirm("¿Eliminar esta tarea?")) return;

    $.ajax({
        url: "/CRM_INT/CRM/controller/TareaController.php?action=delete",
        type: "POST",
        data: { id },
        dataType: "json",
        success: function (res) {
            if (res.success) listarTareas();
            else alert("Error al eliminar la tarea.");
        },
    });
});

// -----
// ◇ Inicializar cargando las tareas
// -----
listarTareas();
});

```

## | └ Usuario.js

```
let usuariosOriginales = [];  
// Nueva variable para guardar TODOS los  
usuarios  
let usuariosActuales = [];  
// Guardará los usuarios filtrados para la  
tabla  
let ultimoUsuarioGuardado = null;  
// Para hacer focus al usuario recién  
guardado  
  
document.addEventListener("DOMContentLoaded", function () {  
    let editandoId = null;  
  
    // Validación de contraseña robusta y detallada  
    function validarContrasena(contrasena) {  
        let errores = [];  
        if (contrasena.length < 6 || contrasena.length > 16) {  
            errores.push("A la contraseña le falta entre 6 y 16 caracteres.");  
        }  
        if (!/[a-z]/.test(contrasena)) {  
            errores.push("A la contraseña le falta al menos una letra  
minúscula.");  
        }  
        if (!/[A-Z]/.test(contrasena)) {  
            errores.push("A la contraseña le falta al menos una letra  
mayúscula.");  
        }  
        if (!/\d/.test(contrasena)) {  
            errores.push("A la contraseña le falta al menos un número.");  
        }  
        if (!/[a-zA-Z\d]/.test(contrasena)) {  
            errores.push("A la contraseña le falta al menos un carácter  
especial.");  
        }  
        return errores;  
    }  
  
    // Validación de usuario completamente en mayúsculas  
    function esUsuarioMayusculas(usuario) {  
        return /^[A-Z]+$/ . test(usuario);  
    }  
  
    const form = document.getElementById("usuarioForm");  
    const formTitulo = document.getElementById("usuarioFormTitulo");  
    const usuarioIdInput = document.getElementById("usuarioId");  
  
    if (form) {  
        form.addEventListener("submit", function (e) {
```

```

e.preventDefault();

const usuario = document.getElementById("usuario").value.trim();
const contrasena =
document.getElementById("contrasena").value.trim();
const rol = document.getElementById("rol").value.trim();
const privilegios =
document.getElementById("privilegios").value.trim();

if (!usuario || (!editandoId && !contrasena) || !rol) {
    alert("Todos los campos obligatorios deben completarse.");
    return;
}

// Nueva validación robusta de contraseña
const errores = validarContrasena(contrasena);
if (!editandoId && errores.length > 0) {
    alert("La contraseña no es válida:\n- " + errores.join("\n- "));
    return;
}
if (editandoId && contrasena && errores.length > 0) {
    alert("La nueva contraseña no es válida:\n- " + errores.join("\n- "));
}
return;

const datos = new FormData();
datos.append("usuario", usuario);
datos.append("rol", rol);
datos.append("privilegios", privilegios);

if (editandoId) {
    datos.append("action", "update");
    datos.append("id", editandoId);
    if (contrasena) datos.append("contrasena", contrasena);
    // Guardar el ID del usuario que se está editando para hacer
focus
    ultimoUsuarioGuardado = editandoId;
} else {
    datos.append("action", "create");
    datos.append("contrasena", contrasena);
    // Para usuarios nuevos, guardaremos el nombre para buscarlo
después
    ultimoUsuarioGuardado = usuario;
}

fetch("/CRM_INT/CRM/controller/UsuarioController.php", {
    method: "POST",

```

```

        body: datos,
    })
    .then((res) => res.json())
    .then((data) => {
        if (data.success) {
            alert(editandoId ? "Usuario editado correctamente." :
"Usuario guardado correctamente.");
            form.reset();
            const esEdicion = editandoId !== null;
            editandoId = null;
            if (formTitulo) formTitulo.textContent = "Registrar Usuario";

            // Recargar usuarios y hacer focus en el usuario
            // guardado/editado
            cargarUsuarios().then(() => {
                if (esEdicion) {
                    // Si fue edición, hacer focus por ID
                    hacerFocusEnUsuario(ultimoUsuarioGuardado, 'id');
                } else {
                    // Si fue creación, hacer focus por nombre de usuario
                    hacerFocusEnUsuario(ultimoUsuarioGuardado, 'usuario');
                }
            });
        } else {
            alert("No se pudo guardar: " + (data.message || "Error
desconocido"));
        }
    })
    .catch((err) => {
        console.error("Error al guardar usuario:", err);
        alert("Error al guardar usuario.");
    });
});
}
}

// Función para hacer focus en un usuario específico de la tabla
function hacerFocusEnUsuario(valor, tipo = 'id') {
    setTimeout(() => {
        let selector;
        if (tipo === 'id') {
            selector = `fila-usuario[data-id="${valor}"]`;
        } else if (tipo === 'usuario') {
            // Buscar por nombre de usuario en los datos
            const usuarioEncontrado = usuariosOriginales.find(u =>
                u.usuario.toLowerCase() === valor.toLowerCase()
            );
            if (usuarioEncontrado) {
                selector = `fila-usuario[data-id="${usuarioEncontrado.id}"]`;
            }
        }
    }, 0);
}

```

```

        }

    }

    if (selector) {
        const filaUsuario = document.querySelector(selector);
        if (filaUsuario) {
            // Hacer scroll hacia el elemento
            filaUsuario.scrollIntoView({
                behavior: 'smooth',
                block: 'center'
            });

            // Agregar clase de highlight temporal
            filaUsuario.classList.add('usuario-destacado');

            // Remover la clase después de 3 segundos
            setTimeout(() => {
                filaUsuario.classList.remove('usuario-destacado');
            }, 3000);
        }
    }
}, 500); // Pequeño delay para asegurar que la tabla se haya
renderizado
}

// Función para llenar el formulario automáticamente
function llenarFormularioAutomatico(usuario) {
    editandoId = usuario.id;

    if (formTitulo) formTitulo.textContent = "Editar Usuario";
    if (usuarioIdInput) usuarioIdInput.value = usuario.id;

    const usuarioInput = document.getElementById("usuario");
    const rolInput = document.getElementById("rol");
    const contrasenaInput = document.getElementById("contrasena");
    const privilegiosInput = document.getElementById("privilegios");

    if (usuarioInput) usuarioInput.value = usuario.usuario;
    if (rolInput) rolInput.value = usuario.rol;
    if (contrasenaInput) contrasenaInput.value = "";

    // Llenar privilegios basado en el rol
    if (privilegiosInput) {
        const mensajesPrivilegios = {
            "Administrador": "Como administrador, tiene acceso completo al
Dashboard. En la sección de Clientes VIP, puede ver, agregar y editar
clientes, pero no puede eliminarlos ni reasignarles tarjetas. En la
sección de Beneficios, puede utilizar todas las funciones, excepto"
        }
    }
}

```

eliminar beneficios ya registrados. No tiene acceso al módulo de Usuarios. En Código de Barras, puede utilizar todas las funcionalidades, incluyendo la exportación a PDF, pero no puede imprimir directamente desde impresora. Tiene acceso completo al módulo de Análisis y a la sección de Cumpleaños, con todas sus funcionalidades. No tiene acceso al módulo de Bitácora.",

"Salonero": "Como salonero, puede ver solo la sección de Tareas en el Dashboard. En la sección de Clientes VIP, solo puede visualizar la información; no puede agregar, editar, eliminar clientes ni reasignar tarjetas. En la sección de Beneficios, puede realizar búsquedas, pero no puede acumular puntos, aplicar descuentos ni eliminar beneficios. No tiene permitido ingresar al módulo de Usuarios. En Código de Barras, tiene permitido realizar búsquedas, actualizar y utilizar la funcionalidad del código de barra, pero no puede exportar a PDF ni imprimir. No tiene permiso de acceso al módulo de Análisis. Tiene acceso completo a Cumpleaños, con todas sus funcionalidades. No tiene acceso permitido a Bitácora.",

"Propietario": "Como propietario, tiene acceso total a todas las funciones y secciones del sistema. No tiene ninguna restricción."

```
};

privilegiosInput.value = mensajesPrivilegios[usuario.rol] || "";

}

// Solo hacer focus en el primer campo sin scroll
if (usuarioInput) {
    setTimeout(() => {
        usuarioInput.focus();
        usuarioInput.select(); // Seleccionar todo el texto para
facilitar edición
    }, 300);
}

document.addEventListener("click", function (e) {
    const editarBtn = e.target.closest(".editar-usuario");
    const eliminarBtn = e.target.closest(".eliminar-usuario");

    // Mensajes de privilegios unificados
    const mensajesPrivilegios = {

        "Administrador": "Como administrador, tiene acceso completo al Dashboard. En la sección de Clientes VIP, puede ver, agregar y editar clientes, pero no puede eliminarlos ni reasignarles tarjetas. En la sección de Beneficios, puede utilizar todas las funciones, excepto eliminar beneficios ya registrados. No tiene acceso al módulo de Usuarios. En Código de Barras, puede utilizar todas las funcionalidades, incluyendo la exportación a PDF, pero no puede imprimir directamente desde impresora. Tiene acceso completo al módulo de Análisis y a la
```

sección de Cumpleaños, con todas sus funcionalidades. No tiene acceso al módulo de Bitácora.",

"Salonero": "Como salonero, puede ver solo la sección de Tareas en el Dashboard. En la sección de Clientes VIP, solo puede visualizar la información; no puede agregar, editar, eliminar clientes ni reasignar tarjetas. En la sección de Beneficios, puede realizar búsquedas, pero no puede acumular puntos, aplicar descuentos ni eliminar beneficios. No tiene permitido ingresar al módulo de Usuarios. En Código de Barras, tiene permitido realizar búsquedas, actualizar y utilizar la funcionalidad del código de barra, pero no puede exportar a PDF ni imprimir. No tiene permiso de acceso al módulo de Análisis. Tiene acceso completo a Cumpleaños, con todas sus funcionalidades. No tiene acceso permitido a Bitácora.",

"Propietario": "Como propietario, tiene acceso total a todas las funciones y secciones del sistema. No tiene ninguna restricción."

};

```
if (editarBtn) {  
    const id = editarBtn.getAttribute("data-id");  
    const usuario = editarBtn.getAttribute("data-usuario");  
    const rol = editarBtn.getAttribute("data-rol");  
    editandoId = id;  
    if (formTitulo) formTitulo.textContent = "Editar Usuario";  
    if (usuarioIdInput) usuarioIdInput.value = id;  
    document.getElementById("usuario").value = usuario;  
    document.getElementById("rol").value = rol;  
    document.getElementById("contrasena").value = "";  
    if (rolInput && privilegiosInput) {  
        privilegiosInput.value = mensajesPrivilegios[rol] || "";  
    }  
    document.getElementById("usuario").focus();  
    return;  
}  
  
if (eliminarBtn) {  
    const id = eliminarBtn.getAttribute("data-id");  
    if (confirm("¿Estás seguro de eliminar este usuario?")) {  
        const datos = new FormData();  
        datos.append("action", "delete");  
        datos.append("id", id);  
  
        fetch("/CRM_INT/CRM/controller/UsuarioController.php", {  
            method: "POST",  
            body: datos,  
        })  
            .then((res) => res.json())  
            .then((data) => {  
                if (data.success) {
```

```

        alert("Usuario eliminado.");
        cargarUsuarios();
    } else {
        alert("No se pudo eliminar: " + (data.message || "Error desconocido."));
    }
})
.catch((err) => {
    console.error("Error:", err);
    alert("Error al eliminar usuario.");
});
}
}
});

const cancelBtn = document.getElementById("cancelBtn");
if (cancelBtn) {
    cancelBtn.addEventListener("click", function () {
        editandoId = null;
        if (formTitulo) formTitulo.textContent = "Registrar Usuario";
        if (form) form.reset();
    });
}

cargarUsuarios();

const usuarioInput = document.getElementById("usuario");
if (usuarioInput) {
    usuarioInput.oninvalid = function (e) {
        e.target.setCustomValidity("Por favor introduzca un nombre");
    };
    usuarioInput.oninput = function (e) {
        e.target.setCustomValidity("");
    };
    usuarioInput.addEventListener("input", function () {
        this.value = this.value.toUpperCase();
    });
}

const passwordInput = document.getElementById("contrasena");
if (passwordInput) {
    passwordInput.setAttribute("maxlength", "16");
    passwordInput.oninvalid = function (e) {
        e.target.setCustomValidity("Por favor introduzca una contraseña");
    };
    passwordInput.oninput = function (e) {
        e.target.setCustomValidity("");
    };
}

```

```

}

const rolInput = document.getElementById("rol");
const privilegiosInput = document.getElementById("privilegios");

if (rolInput) {
    rolInput.oninvalid = function (e) {
        e.target.setCustomValidity("Por favor seleccione un rol");
    };
    rolInput.oninput = function (e) {
        e.target.setCustomValidity("");
    };
}

if (rolInput && privilegiosInput) {
    rolInput.addEventListener("change", function () {
        const mensajesPrivilegios = {
            "Administrador": "Como administrador, tiene acceso completo al Dashboard. En la sección de Clientes VIP, puede ver, agregar y editar clientes, pero no puede eliminarlos ni reasignarles tarjetas. En la sección de Beneficios, puede utilizar todas las funciones, excepto eliminar beneficios ya registrados. No tiene acceso al módulo de Usuarios. En Código de Barras, puede utilizar todas las funcionalidades, incluyendo la exportación a PDF, pero no puede imprimir directamente desde impresora. Tiene acceso completo al módulo de Análisis y a la sección de Cumpleaños, con todas sus funcionalidades. No tiene acceso al módulo de Bitácora.",

            "Salonero": "Como salonero, puede ver solo la sección de Tareas en el Dashboard. En la sección de Clientes VIP, solo puede visualizar la información; no puede agregar, editar, eliminar clientes ni reasignar tarjetas. En la sección de Beneficios, puede realizar búsquedas, pero no puede acumular puntos, aplicar descuentos ni eliminar beneficios. No tiene permitido ingresar al módulo de Usuarios. En Código de Barras, tiene permitido realizar búsquedas, actualizar y utilizar la funcionalidad del código de barra, pero no puede exportar a PDF ni imprimir. No tiene permiso de acceso al módulo de Análisis. Tiene acceso completo a Cumpleaños, con todas sus funcionalidades. No tiene acceso permitido a Bitácora.",

            "Propietario": "Como propietario, tiene acceso total a todas las funciones y secciones del sistema. No tiene ninguna restricción."
        };
        privilegiosInput.value = mensajesPrivilegios[rolInput.value] || "";
    });
}

if (privilegiosInput) {
    privilegiosInput.value = "";
}

```

```

});;

//LÓGICA PARA EL BUSCADOR Y LA FORMA EN QUE SE MUESTRAN LOS USUARIOS
//ADEMÁS DEL AUTOCOMPLETADO DEL FORMULARIO
function mostrarUsuarios(usuarios) {
    usuariosActuales = usuarios; // Guardar para eventos de la tabla

    const contenedor = document.getElementById("usuarioLista");
    if (!usuarios || usuarios.length === 0) {
        contenedor.innerHTML = "<p class='text-center mt-3'>No hay usuarios registrados.</p>";
        return;
    }

    let html = `<table class='table table-striped table-hover mt-4' id='tablaUsuarios'>
        <thead class='table-dark'>
            <tr>
                <th>ID</th>
                <th>Usuario</th>
                <th>Rol</th>
                <th>Acciones</th>
            </tr>
        </thead>
        <tbody>`;

    usuarios.forEach(usuario => {
        html += `<tr class="fila-usuario" data-id="${usuario.id}" style="cursor: pointer;">
            <td>${usuario.id}</td>
            <td>${usuario.usuario}</td>
            <td>${usuario.rol.toUpperCase()}</td>
            <td>
                <button class='btn btn-sm btn-warning editar-usuario' data-id='${usuario.id}' data-usuario='${usuario.usuario}' data-rol='${usuario.rol}'>Editar</button>
                <button class='btn btn-sm btn-danger eliminar-usuario' data-id='${usuario.id}'>Eliminar</button>
            </td>
        </tr>`;
    });

    html += "</tbody></table>";
    contenedor.innerHTML = html;

    // --- Auto-llenado de formulario al dar click en la fila (NO en los botones)
    document.querySelectorAll(".fila-usuario").forEach(fila => {

```

```

fila.addEventListener("click", function(e) {
    // Prevenir si se hace clic en botones
    if (e.target.tagName === "BUTTON" || e.target.closest("button"))
return;

    const id = this.getAttribute("data-id");

    // Buscar en usuariosOriginales el usuario correspondiente
    const usuario = usuariosOriginales.find(u => u.id == id);

    if (usuario) {
        llenarFormularioAutomatico(usuario);
    }
});

// Agregar efecto hover visual para indicar que es clickeable
fila.addEventListener("mouseenter", function(e) {
    if (!e.target.closest("button")) {
        this.style.backgroundColor = "#f8f9fa";
    }
});

fila.addEventListener("mouseleave", function() {
    this.style.backgroundColor = "";
});

});

// Función para llenar el formulario automáticamente
function llenarFormularioAutomatico(usuario) {
    editandoId = usuario.id;

    const usuarioIdInput = document.getElementById("usuarioId");
    if (usuarioIdInput) usuarioIdInput.value = usuario.id;

    const usuarioInput = document.getElementById("usuario");
    const rolInput = document.getElementById("rol");
    const contrasenaInput = document.getElementById("contrasena");
    const privilegiosInput = document.getElementById("privilegios");

    if (usuarioInput) usuarioInput.value = usuario.usuario;
    if (rolInput) rolInput.value = usuario.rol;
    if (contrasenaInput) contrasenaInput.value = "";

    // Llenar privilegios basado en el rol
    if (privilegiosInput) {
        const mensajesPrivilegios = {
            "Administrador": "...",

```

```

        "Salonero": "...",
        "Propietario": "..."
    };
    privilegiosInput.value = mensajesPrivilegios[usuario.rol] || "";
}

// Solo hacer focus en el primer campo sin scroll
if (usuarioInput) {
    setTimeout(() => {
        usuarioInput.focus();
        usuarioInput.select(); // Seleccionar todo el texto para facilitar
edición
    }, 300);
}

// Función para inicializar o actualizar la lista completa de usuarios
const cargarUsuariosCompletos = (usuarios) => {
    usuariosOriginales = [...usuarios].sort((a, b) => b.id - a.id); // Más
reciente primero
    mostrarUsuarios(usuariosOriginales);
};

window.cargarUsuarios = function () {
    const contenedor = document.getElementById("usuarioLista");
    if (!contenedor) return Promise.reject("Contenedor no encontrado");

    contenedor.innerHTML = `<div class="text-center p-3">
        <div class="spinner-border text-primary" role="status"></div>
        <p class="mt-2">Cargando usuarios...</p>
    </div>`;

    return
fetch("/CRM_INT/CRM/controller/UsuarioController.php?action=readAll")
    .then((res) => res.json())
    .then((data) => {
        if (data.success && Array.isArray(data.data)) {
            // CAMBIO IMPORTANTE: Usar la nueva función en lugar de
mostrarUsuarios directamente
            cargarUsuariosCompletos(data.data);
            return data.data; // Retornar los datos para poder usar en el
promise
        } else {
            contenedor.innerHTML = "<p class='text-danger'>Error al cargar
usuarios.</p>";
            throw new Error("Error al cargar usuarios");
        }
    })
}

```

```

        })
        .catch((err) => {
            console.error("Error al cargar usuarios:", err);
            contenedor.innerHTML = "<p class='text-danger'>Error al cargar
usuarios.</p>";
            throw err;
        });
    });

const buscadorUsuarios = document.getElementById("buscadorUsuarios");
if (buscadorUsuarios) {
    buscadorUsuarios.addEventListener("input", function () {
        const valor = this.value.trim().toLowerCase();

        if (!valor) {
            // Si usuariosOriginales está vacío, usar usuariosActuales como
fallback
            const usuariosParaMostrar = usuariosOriginales.length > 0 ?
usuariosOriginales : usuariosActuales;
            mostrarUsuarios(usuariosParaMostrar);
            return;
        }

        // Usar el mismo fallback para filtrar
        const usuariosParaFiltrar = usuariosOriginales.length > 0 ?
usuariosOriginales : usuariosActuales;
        const filtrados = usuariosParaFiltrar.filter(u =>
            (u.usuario && u.usuario.toLowerCase().includes(valor)) ||
            (u.id && u.id.toString().includes(valor))
        );

        mostrarUsuarios(filtrados);
    });
}

document.getElementById("usuarioForm").addEventListener("keydown",
function(e) {
    // Solo para Enter, no estando en textarea ni en botón de cancelar
    if (
        e.key === "Enter" &&
        e.target.tagName !== "TEXTAREA" &&
        !(e.target.id === "cancelBtn")
    ) {
        e.preventDefault(); // Evita el submit por defecto (doble envío)
        this.requestSubmit(); // Lanza el submit (respeta validaciones)
    }
});

```

```

// Función para hacer focus en un usuario específico de la tabla
function hacerFocusEnUsuario(valor, tipo = 'id') {
    setTimeout(() => {
        let selector;
        if (tipo === 'id') {
            selector = `fila-usuario[data-id="${valor}"]`;
        } else if (tipo === 'usuario') {
            // Buscar por nombre de usuario en los datos
            const usuarioEncontrado = usuariosOriginales.find(u =>
                u.usuario.toLowerCase() === valor.toLowerCase()
            );
            if (usuarioEncontrado) {
                selector = `fila-usuario[data-id="${usuarioEncontrado.id}"]`;
            }
        }
        if (selector) {
            const filaUsuario = document.querySelector(selector);
            if (filaUsuario) {
                // Hacer scroll hacia el elemento
                filaUsuario.scrollIntoView({
                    behavior: 'smooth',
                    block: 'center'
                });

                // Agregar clase de highlight temporal
                filaUsuario.classList.add('usuario-destacado');

                // Remover la clase después de 3 segundos
                setTimeout(() => {
                    filaUsuario.classList.remove('usuario-destacado');
                }, 3000);
            }
        }
    }, 500); // Pequeño delay para asegurar que la tabla se haya
    renderizado
}

```

```

|__ view/
|  |__ AnalisisView.php
<div class="card-header bg-dark text-white rounded-top p-4">
    <!-- Título (línea 1) -->
    <div class="d-flex align-items-center mb-2"> <!-- <== AJUSTA ESTE mb-2 ó mb-1 -->
        <h5 class="mb-0 header-title">
            <i class="fa-solid fa-chart-bar"></i> Análisis de Clientes

```

```

        </h5>
        <div class="flex-grow-1 ms-3 divider-header"></div>
    </div>

    <!-- Fila 1 de botones -->
    <div class="d-flex flex-wrap gap-2 w-100 botones-analisis mb-2">
        <button class="btn btn-analisis active flex-fill"
id="btnClientesFrecuentes">Clientes más frecuentes</button>
        <button class="btn btn-analisis flex-fill"
id="btnClientesMayorHistorial">
            <i class="fa-solid fa-crown"></i> Clientes con mayor
historial
        </button>
        <button class="btn btn-analisis flex-fill"
id="btnClientesInactivos">Clientes inactivos</button>
        <button class="btn btn-analisis flex-fill"
id="btnResidenciasFrecuentes">Residencias más frecuentes</button>
    </div>
    <!-- Fila 2 de botones + buscador -->
    <div class="d-flex flex-wrap gap-2 w-100 align-items-center">
        <button class="btn btn-analisis btn-outline-dark flex-fill"
id="btnClientesAntiguos">
            Clientes más antiguos
        </button>
        <button class="btn btn-analisis btn-outline-dark flex-fill"
id="btnVentasPorMes">Ventas por mes</button>
        <button class="btn btn-analisis flex-fill" id="btnVentasPorAnio">
            <i class="fa-solid fa-calendar-alt"></i> Ventas por año
        </button>
        <button class="btn btn-analisis btn-outline-white flex-fill"
id="btnActualizarAnalisis">
            <i class="fa-solid fa-rotate"></i> Actualizar
        </button>
        <!-- Buscador: siempre alineado a la derecha -->
        <div class="input-group buscador-analisis ms-auto" style="min-
width:220px;max-width:300px;">
            <span class="input-group-text bg-dark text-white border-0
rounded-start px-3" id="labelBuscador">
                <i class="fa fa-search"></i>
            </span>
            <input type="text" class="form-control form-control-sm
border-0 rounded-end" placeholder="Buscar..." id="analisisBuscadorGeneral" autocomplete="off" aria-
label="Buscar"
                style="background:#fff; color:#222; font-weight:500;">
        </div>
    </div>
</div>

```

```

<div class="card-body p-0 bg-white rounded-bottom">
    <div class="table-responsive px-4 pt-4 pb-2" id="analisisTablaCont">
        <!-- Aquí irá la tabla dinámica -->
    </div>
    <div class="px-4 pb-4" id="analisisGraficoCont" style="min-height:300px;">
        <!-- Aquí irá el gráfico dinámico -->
    </div>
</div>
</div>
<link rel="stylesheet" href="/CRM_INT/CRM/public/css/Analisis.css?v=<?=time() ?>">
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script src="/CRM_INT/CRM/public/js/Analisis.js?v=<?=time() ?>"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

```

## BitacoraView.php

```

<?php
// Llama al endpoint del controlador para conocer si debe mostrarse
// la alerta preventiva (registros por expirar en ~5 días).
// Sugerencia: reemplazar por cURL con timeout configurable.
$expiracionJson =
file_get_contents('http://localhost/CRM_INT/CRM/controller/BitacoraControler.php?action=verificarExpiracion');
$mostrarAlerta = json_decode($expiracionJson, true)['mostrarAlerta'] ?? false;
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Bitácora</title>
    <!-- Iconos -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/6.0.0/css/all.min.css">
    <!-- Estilos del módulo (cache-busting de desarrollo) -->
    <link rel="stylesheet" href="/CRM_INT/CRM/public/css/Bitacora.css?v=<?=time() ?>">
</head>

```

```

<body>
    <!-- aquí va el texto fijo que siempre se muestra, avisando que los
registros se borran a los 30 días. -->
    <div class="alert">
        Los registros se eliminan automáticamente al cumplir 30 días.
        Puede exportar un respaldo en PDF o Excel antes de esa fecha.
    </div>

    <?php if ($mostrarAlerta): ?>
        <!-- Alerta preventiva condicional (5 días antes) -->
        <div class="alert">
            <i class="fas fa-exclamation-triangle"></i>
            Algunos registros serán eliminados en 5 días. Se recomienda
realizar un respaldo.
        </div>
    <?php endif; ?>

    <!-- Acciones de exportación a Pdf y Excell-->
    <button class="btn btn-pdf" onclick="exportToPDF()">Exportar
PDF</button>
    <button class="btn btn-excel" onclick="exportToExcel()">Exportar
Excel</button>

    <!-- en este div (una caja vacía en HTML) el archivo Bitacora.js
se encarga de poner la tabla con los registros cuando se cargue la
página. -->
    <div id="contenedorTabla" class="table-container"></div>

    <!-- Librerías necesarias para PDF -->
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"
></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf-
autotable/3.5.28/jspdf.plugin.autotable.min.js"></script>

    <!-- Lógica del módulo -->
    <script src="/CRM_INT/CRM/public/js/Bitacora.js?v=<?= time()
?>"></script>
</body>
</html>

```

## | └ ClientView.php

```

<!-- Líneas de adorno visual en la página -->
<div class="decor-line" style="top: 120px; left: 40%;"></div>
<div class="decor-line" style="top: 250px; left: 20%; width:
80px;"></div>

```

```

<div class="container mt-4 mover-derecha">

    <!-- Tarjeta con el formulario para registrar o editar un cliente -->
    <div class="card shadow mb-4">
        <div class="card-header clientes-header d-flex align-items-center">
            <h5 class="mb-0 flex-grow-1 titulo-clientes">Registro de Cliente</h5>
        </div>
        <div class="card-body">
            <form id="clienteForm">
                <div class="row">

                    <!-- Campo ID: se llena automático, no editable -->
                    <div class="col-md-4 mb-3">
                        <label for="clienteId" class="form-label">ID</label>
                        <input type="text" name="id" id="id" class="form-control" placeholder="Autogenerado" readonly>
                    </div>

                    <!-- Campo para la cédula: acepta 9 o 14 dígitos -->
                    <div class="col-md-4 mb-3">
                        <label for="clienteCedula" class="form-label">Cédula *</label>
                        <input type="text" name="cedula" id="clienteCedula" class="form-control" required
                               placeholder="Ej: 123456789" maxlength="14"
                               pattern="\d{9}|\d{14}">
                    </div>

                    <!-- Campo para el nombre completo -->
                    <div class="col-md-4 mb-3">
                        <label for="clienteNombre" class="form-label">Nombre Completo *</label>
                        <input type="text" name="nombre" id="clienteNombre" class="form-control" required maxlength="45"
                               placeholder="Ej: JUAN PÉREZ GONZÁLEZ">
                    </div>
                </div>
            </div>

            <div class="row">
                <!-- Campo de correo electrónico -->
                <div class="col-md-6 mb-3">

```

```

        <label for="clienteCorreo" class="form-label">Correo Electrónico *</label>
                    <input type="email" name="correo"
id="clienteCorreo" class="form-control"
placeholder="ejemplo@correo.com"
maxlength="30">
                </div>

                    <!-- Campo de teléfono -->
<div class="col-md-6 mb-3">
    <label for="clienteTelefono" class="form-label">Teléfono *</label>
        <input type="tel" name="telefono"
id="clienteTelefono" class="form-control" required
placeholder="0000-0000">
    </div>
</div>

<div class="row">

    <!-- Campo de lugar de residencia, usa lista de cantones
-->
    <div class="col-md-6 mb-3">
        <label for="clienteLugar" class="form-label">Lugar de Residencia *</label>
            <input list="listaCantones"
name="lugarResidencia" id="clienteLugar" class="form-control"
required placeholder="Ej: San José, Costa Rica">
            <datalist id="listaCantones"></datalist>
    </div>

    <!-- Campo de cumpleaños con validación de fechas
-->
    <div class="col-md-6 mb-3">
        <label for="clienteFecha" class="form-label">Fecha de Cumpleaños *</label>
        <input
            type="date"
            name="fechaCumpleanos"
            id="clienteFecha"
            class="form-control"
            required
            min="1900-01-01"
            max="<?= date('Y-m-d') ?">
        >
    </div>

```

```

        </div>
        <div class="row">

            <!-- Campo para alergias del cliente -->

            <div class="col-md-6 mb-3">
                <label for="clienteAlergias" class="form-label">Alergias</label>
                    <textarea name="alergias" id="clienteAlergias" class="form-control" rows="2" maxlength="255"
                        placeholder="Ej: Maní, lácteos,
gluten..."></textarea>
            </div>

            <!-- Campo para gustos especiales del cliente -->
            <div class="col-md-6 mb-3">
                <label for="clienteGustos" class="form-label">Gustos Especiales</label>
                    <textarea name="gustosEspeciales" id="clienteGustos" class="form-control" rows="2"
                        maxlength="255"
                        placeholder="Ej: Prefiere platillos sin
picante, jugos naturales..."></textarea>
            </div>
        </div>

        <!-- Botones para guardar o cancelar -->
        <div class="d-flex justify-content-end">
            <button type="button" id="cancelBtn" class="btn btn-secondary me-2"
                onclick="cancelarEdicion()">Cancelar</button>
            <button type="submit" id="submitBtn" class="btn btn-primary">Guardar Cliente</button>
        </div>
    </form>
</div>
</div>

        <!-- Tarjeta con la lista de clientes ya registrados -->
<div class="card shadow mb-4">
    <div class="card-header clientes-header d-flex align-items-center
gap-3 flex-wrap">
        <h5 class="mb-0 flex-grow-1 titulo-clientes">Lista de
Clientes</h5>
        <!-- Buscador en tiempo real -->
        <input type="text" id="buscadorClientes" class="form-control
buscador-clientes">
    </div>

```

```

        placeholder="Buscar por nº de tarjeta o nombre...""
autocomplete="off">
    <button class="btn btn-sm btn-outline-light ms-2"
onclick="cargarClientes()">
        Actualizar
    </button>
</div>
<div class="card-body p-0">
    <div class="table-responsive">
        <div id="clienteLista" class="p-4 text-center">
            <!-- Aquí JS inyecta la tabla de clientes -->
        </div>
    </div>
</div>
</div>

<!-- Tarjeta con el historial de reasignaciones de tarjetas VIP -->
<div class="card shadow mb-4">
    <div class="card-header clientes-header d-flex align-items-center
gap-3 flex-wrap">
        <h5 class="mb-0 flex-grow-1 titulo-clientes">
            <i class="fas fa-history me-2"></i>Historial de
Reasignaciones
        </h5>
        <small class="text-light me-2">
            <i class="fas fa-sync-alt me-1"></i>
            <span id="estadoActualizacion">Actualización automática
activa</span>
        </small>
            <!-- Botón para actualizar el historial manualmente -->
            <button class="btn btn-sm btn-outline-light ms-2"
onclick="cargarHistorialReasignaciones(true)">
                title="Actualizar manualmente">
                <i class="fas fa-refresh me-1"></i>Actualizar
            </button>
        </div>
        <div class="card-body p-0">
            <div class="table-responsive">
                <div id="historialReasignaciones" class="p-4 text-
center">
                    <!-- Aquí se inyectará la tabla del historial -->
                </div>
            </div>
        </div>
    </div>
</div>

<!-- Modal para reasignación de código -->

```

```

<div class="modal fade" id="modalReasignar" tabindex="-1" aria-
labelledby="modalReasignarLabel" aria-hidden="true"
    data-bs-backdrop="false" data-bs-keyboard="true">
    <div class="modal-dialog modal-dialog-centered">
        <div class="modal-content">
            <div class="modal-header reasignar-header">
                <h6 class="modal-title" id="modalReasignarLabel">
                    <i class="fas fa-sync-alt me-2"></i>Reasignar

```

Código

```

                </h6>
                <button type="button" class="btn-close btn-close-
white" data-bs-dismiss="modal"
                    aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <!-- Datos del cliente en la ventana -->
                <div class="cliente-info-card-compact">
                    <div class="text-center mb-3">
                        <h6 class="mb-2"><i class="fas fa-user me-
2"></i><span id="modalClienteNombre">-</span></h6>
                        <div class="cliente-datos-inline">
                            <span class="badge bg-secondary me-2">ID:<
span id="modalClienteId">-</span></span>
                            <span class="badge bg-secondary">Cédula:<
span id="modalClienteCedula">-</span></span>
                        </div>
                    </div>
                    <!-- Motivo de la reasignación -->
                    <div class="mt-3">
                        <label for="motivoReasignacion" class="form-
label">
                            <strong><i class="fas fa-edit me-
1"></i>Motivo:</strong> *
                        </label>
                        <select class="form-select mb-2"
id="motivoSelect" onchange="manejarSeleccionMotivo()">
                            <option value="">Seleccionar
motivo...</option>
                            <option value="Tarjeta dañada">Tarjeta
dañada</option>
                            <option value="Tarjeta perdida">Tarjeta
perdida</option>
                            <option value="Tarjeta robada">Tarjeta
robada</option>
                            <option value="Tarjeta deteriorada">Tarjeta
deteriorada</option>

```

```

                <option value="Reemplazo por
desgaste">Reemplazo por desgaste</option>
                <option value="Solicitud del
cliente">Solicitud del cliente</option>
                <option value="Otro">Otro
(especificar)</option>
            </select>
            <textarea class="form-control motivo-textarea-
compact" id="motivoReasignacion" rows="2"
placeholder="Descripción del motivo...">
            <div class="form-text">Máximo 100
caracteres</div>
        </div>

        <!-- Mensaje de información del proceso -->
        <div class="alert alert-info alert-compact mt-3"
role="alert">
            <i class="fas fa-info-circle me-2"></i>
            <small><strong>Proceso:</strong> Se generará un
nuevo código VIP automáticamente.</small>
        </div>
    </div>

        <!-- Botones del modal -->
        <div class="modal-footer justify-content-center">
            <button type="button" class="btn btn-secondary btn-
sm" data-bs-dismiss="modal">
                <i class="fas fa-times me-1"></i>Cancelar
            </button>
            <button type="button" class="btn btn-warning btn-sm"
id="btnConfirmarReasignacion"
                onclick="procesarReasignacion()">
                <i class="fas fa-sync-alt me-1"></i>Confirmar
            </button>
        </div>
    </div>
</div>

<!-- Script exclusivo de esta vista -->
<script src="/CRM_INT/CRM/public/js/Cliente.js?v=<?= time()
?>"></script>
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
```

## | └-- CódigoView.php

```
<!-- Tarjeta que contiene la lista de tarjetas -->
<div class="card shadow mb-4">

    <!-- Encabezado con título y acciones -->
    <div class="card-header d-flex justify-content-between align-items-center bg-dark text-white">
        <h5 class="mb-0">Lista de Tarjetas</h5>
        <div class="d-flex gap-2">
            <!-- Campo de texto para buscar por número de tarjeta o nombre -->
            <input type="text" id="buscarCodigo" class="form-control form-control-sm w-auto" placeholder="Nº tarjeta o Nombre">

            <!-- Botón para ejecutar la búsqueda -->
            <button id="btnBuscarTarjeta" class="btn btn-sm btn-warning" title="Ir a Beneficios">
                <i class="bi bi-search"></i> Buscar
            </button>

            <!-- Botón para recargar/actualizar la lista -->
            <button id="btnActualizar" class="btn btn-sm btn-outline-light">
                Actualizar
            </button>
        </div>
    </div>

    <!-- Cuerpo donde se mostrará la tabla -->
    <div class="card-body p-0">
        <div class="table-responsive">
            <div id="codigoLista" class="p-4">
                <!-- Aquí el JS dibuja la tabla de tarjetas -->
            </div>
        </div>
    </div>
</div>

<!-- Script propio que maneja la lógica de tarjetas -->
<script src="/CRM_INT/CRM/public/js/Codigo.js?v=<?= time() ?>"></script>

<!-- Librería para generar códigos de barras -->
```

```

<script
src="https://cdn.jsdelivr.net/npm/jsbarcode@3.11.5/dist/JsBarcode.all.min.js"></script>


<script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"
></script>


<script src="https://cdn.jsdelivr.net/npm/canvas-confetti@1.6.0/dist/confetti.browser.min.js"></script>


<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

```

## | └ CompraView.php

```

<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CompraView</title>
    <!-- Estilos propios de esta vista -->
    <link rel="stylesheet" href="/CRM_INT/CRM/public/css/Compra.css">
    <!-- Íconos de Bootstrap -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.5/font/bootstrap-icons.css">
</head>

<body>
    <div class="compra-layout">
        <!-- Barra decorativa amarilla superior -->
        <div class="compra-barra-amarilla"></div>
        <div class="template-container">

            <!-- Bloque principal donde se selecciona la acción -->
            <div class="compra-bloque-inicio">
                <div class="compra-header-linea">
                    <!-- Mensaje de error dinámico -->
                    <div id="compraMensaje" class="mensaje-error"></div>

                    <!-- Título y botones de acción -->
                    <div class="compra-titulo">Elije una opción</div>

```

```

        <button class="compra-btn-opcion"
id="compraOpcionCompra">Compra</button>
            <button class="compra-btn-opcion"
id="compraOpcionDescuento">Descuento</button>
            <button class="compra-btn-opcion"
id="compraOpcionExpress" style="display:none;">Express</button>

        <!-- Input para buscar cliente por número de tarjeta
-->
        <div class="compra-input-container">
            <input type="number" class="compra-input-id"
id="compraInputId" placeholder="# de Tarjeta">
                <!-- Botón de búsqueda rápido (solo se muestra en
modo "Acumular") -->
                <button class="compra-btn-buscar-icon"
id="compraBtnBuscarIcon" style="display: none;">
                    title="Buscar otra tarjeta">
                    <i class="bi bi-search"></i>
                </button>
            </div>
            <!-- Botón para buscar la tarjeta -->
            <button class="compra-btn-buscar"
id="compraBtnBuscar">Buscar</button>
        </div>

        <!-- Texto de ayuda para el usuario -->
        <div class="compra-indicacion" id="compraIndicacion">
            <i class="bi bi-info-circle"></i>
            Selecciona una opción para aplicar una compra o
descuento al cliente.
        </div>

        <!-- Formulario con los datos del cliente (se muestra
solo cuando hay datos) -->
        <div class="card shadow mb-4" id="compraCardForm"
style="display:none;">
            <div class="card-body">
                <form id="compraForm">
                    <div class="row">
                        <!-- Datos básicos del cliente -->
                        <div class="col-md-4 mb-3">
                            <label class="form-label">ID</label>
                            <input type="text" id="id"
class="form-control" readonly>
                        </div>
                        <div class="col-md-4 mb-3">
                            <label class="form-
label">Cédula</label>

```

```

        <input type="text" id="clienteCedula"
class="form-control" readonly>
    </div>
    <div class="col-md-4 mb-3">
        <label class="form-label">Nombre
Completo</label>
        <input type="text" id="clienteNombre"
class="form-control" readonly>
    </div>
    </div>
    <div class="row">
        <div class="col-md-6 mb-3">
            <label class="form-
label">Correo</label>
            <input type="email"
id="clienteCorreo" class="form-control" readonly>
        </div>
        <div class="col-md-6 mb-3">
            <label class="form-
label">Teléfono</label>
            <input type="tel"
id="clienteTelefono" class="form-control" readonly>
        </div>
        </div>
        <div class="row">
            <div class="col-md-6 mb-3">
                <label class="form-label">Lugar de
Residencia</label>
                <input type="text" id="clienteLugar"
class="form-control" readonly>
            </div>
            <div class="col-md-6 mb-3">
                <label class="form-label">Fecha de
Cumpleaños</label>
                <input type="date" id="clienteFecha"
class="form-control" readonly>
            </div>
            </div>
            <div class="row">
                <!-- Campo para acumular una compra -->
                <div class="col-md-6 mb-3">
                    <label class="form-label">Acumular
Compra</label>
                    <input type="text"
id="cantidadAcumulada" class="form-control">
                </div>
                <!-- Total acumulado actual del cliente -
->

```

```

        <div class="col-md-6 mb-3">
            <label class="form-label">Total
Actual</label>
            <input type="number" id="totalActual"
class="form-control" readonly>
        </div>
    </div>
</div>

        <!-- Sección para mostrar el historial de compras del
cliente -->
        <div id="historialCompras" style="margin-top: -1.5em;
width:100% ;

height: 100px; min-height: 100px;
overflow-y: auto; overflow-x: auto;
scrollbar-width: none;
-ms-overflow-style: none;
border-radius: 0.5em;">
    </div>
</div>
</div>

        <!-- Barra decorativa lateral derecha -->
        <div class="compra-barra-derecha"></div>
</div>

        <!-- Modal que aparece cuando un cliente está de cumpleaños -->
        <div class="modal fade" id="modalCumpleanos" tabindex="-1" aria-
labelledby="modalCumpleanosLabel"
aria-hidden="true">
        <div class="modal-dialog modal-dialog-centered">
            <div class="modal-content" style="border: 3px solid #f9c41f;
border-radius: 15px;">
                <div class="modal-header"
                    style="background: linear-gradient(135deg, #f9c41f
0%, #e6b619 100%); color: #000;">
                    <h5 class="modal-title fw-bold"
id="modalCumpleanosLabel">
                        🎉 ¡CUMPLEAÑOS ESPECIAL! 🎉
                    </h5>
                </div>
                <div class="modal-body text-center p-4">
                    <div class="mb-3">
                        <i class="fas fa-birthday-cake" style="font-size:
3rem; color: #f9c41f;"></i>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        <!-- Nombre del cumpleañero -->
        <h4 class="fw-bold mb-3" id="nombreCumpleanero"
style="color: #000;"></h4>
            <p class="lead mb-4" style="color: #555;">
                ¡Está celebrando su cumpleaños esta semana! 🎉
            </p>
            <div class="alert alert-warning" role="alert">
                <i class="fas fa-gift me-2"></i>
                <strong>¡Recordale que puede reclamar su regalo
especial!</strong>
            </div>
        </div>
        <div class="modal-footer justify-content-center"
style="border-top: 2px solid #f9c41f;">
            <button type="button" class="btn btn-secondary" data-
bs-dismiss="modal">
                <i class="fas fa-times"></i> Cerrar
            </button>
            <!-- Botón para ir a la sección de cumpleaños -->
            <button type="button" class="btn btn-primary"
id="btnIrCumpleanos"
style="background: #f9c41f; border-color:
#f9c41f; color: #000; font-weight: bold;">
                <i class="fas fa-birthday-cake"></i> Ir a
Cumpleaños
            </button>
        </div>
    </div>
</div>

<!-- Modal Express para mandar correo rápido a un cliente -->
<div class="modal fade" id="modalExpress" tabindex="-1" aria-
hidden="true">
    <div class="modal-dialog modal-dialog-centered">
        <div class="modal-content" style="border:3px solid var(--amarillo);border-radius:15px;">
            <div class="modal-header"
style="background:linear-gradient(135deg,#f9c41f 0%,#e6b619 100%);color:#000;">
                <h5 class="modal-title fw-bold">✉ EXPRESS</h5>
            </div>

            <div class="modal-body text-center p-4">
                <p class="lead mb-3">Estás por usar el apartado
<strong>Express</strong></p>
                <p><strong>Cliente:</strong> <span
id="expressNombre"></span></p>

```

```

        <p><strong>Correo:</strong> <span
id="expressCorreο"></span></p>
        </div>

        <div class="modal-footer justify-content-center"
style="border-top:2px solid var(--amarillo);">
            <!-- Botón que dispara el envío de correo -->
            <button type="button" class="btn btn-primary"
id="btnEnviarCorreoExpress"
                style="background:var(--amarillo);border-
color:var(--amarillo);color:#000;font-weight:bold;">
                <i class="bi bi-envelope-fill"></i> Enviar correo
            </button>
        </div>
    </div>
</div>

<!-- Scripts usados en esta vista -->
<!-- Confeti para efectos visuales -->
<script src="https://cdn.jsdelivr.net/npm/canvas-
confetti@1.6.0/dist/confetti.browser.min.js"></script>
<!-- Lógica propia de compras -->
<script src="/CRM_INT/CRM/public/js/Compra.js"></script>
<!-- Iconos de FontAwesome -->
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css">
</body>

</html>

```

## | └— CumpleView.php

```

<!-- CUMPLEAÑOS DE LA SEMANA -->
<div class="card shadow mb-4">
    <!-- Encabezado con título y botones para cambiar de semana -->
    <div class="card-header d-flex justify-content-between align-items-
center bg-dark text-white">
        <h5 class="mb-0">Cumpleaños de la Semana</h5>
        <div class="d-flex align-items-center gap-2">
            <!-- Botón semana anterior -->
            <button id="btnPrevSemana" class="btn-nav-week" title="Semana
actual" disabled>&lsh;</button>
            <!-- Texto con el rango de la semana -->
            <span id="lblSemana" class="badge bg-warning text-
dark">SEMANA ACTUAL</span>
            <!-- Botón semana siguiente -->

```

```

        <button id="btnNextSemana" class="btn-nav-week" title="Semana
siguiente">&rsaquo;</button>
    </div>

</div>

<div class="card-body p-0">
    <div class="table-responsive">
        <!-- Aquí se mostrará el rango de fechas de la semana -->
        <div id="rangoSemana" class="alert alert-info text-center fw-
bold m-4"></div>

        <!-- Aquí JS inyectará la tabla de clientes con cumpleaños -->
        <div id="cumpleLista" class="p-4"></div>

        <!-- RECORDATORIO DE LLAMADAS -->
        <div id="recordatorioLlamadas" class="mt-2 mx-4 mb-4 d-none"
hidden>

            <ul id="listaRecordatorios" class="mb-0 ps-3 small"
hidden></ul>

            </div>
        </div>
    </div>
</div>

<!-- FORMULARIO DE ENVÍO DE CORREO -->
<div class="card shadow-sm mt-3" style="border-left: 5px solid #f9c41f;">
    <div class="card-header bg-black text-warning py-2">
        <h6 class="mb-0"><i class="fas fa-envelope"></i> Enviar correo de
felicitación</h6>
    </div>
    <div class="card-body p-3">
        <form id="formCorreo" class="row g-2">
            <!-- ID oculto del cumpleañero -->
            <input type="hidden" id="idCumple">

            <!-- Datos del cliente que cumple años -->
            <div class="col-12 col-md-6">
                <label class="form-label small">Nombre</label>
                <input type="text" id="nombreCorreo" class="form-control
form-control-sm" readonly>
            </div>

            <div class="col-6 col-md-3">
                <label class="form-label small">Cédula</label>

```

```

        <input type="text" id="cedulaCorreo" class="form-control form-control-sm" readonly>
    </div>

    <div class="col-6 col-md-3">
        <label class="form-label small">Teléfono</label>
        <input type="text" id="telefonoCorreo" class="form-control form-control-sm" readonly>
    </div>

    <div class="col-12">
        <label class="form-label small">Correo</label>
        <input type="email" id="correoCorreo" class="form-control form-control-sm" readonly>
    </div>

    <!-- Campos ocultos para mensaje y fecha -->
    <input type="hidden" id="mensajeCorreo">
    <input type="hidden" id="fechaCumple">

    <!-- Botones para enviar WhatsApp o Correo -->
    <div class="col-12 d-flex justify-content-end gap-2 mt-2">
        <button id="btnEnviarWhats" type="button" class="btn btn-sm btn-success">
            <i class="bi bi-whatsapp"></i> Enviar WhatsApp
        </button>
        <button id="btnEnviarCorreo" type="submit" class="btn btn-sm btn-warning text-black">
            <i class="fas fa-paper-plane"></i> Enviar correo
        </button>
    </div>
</form>
</div>
</div>

<!-- HISTORIAL DE ENVÍOS -->
<div class="card shadow-sm mt-3">
    <div class="card-header bg-secondary text-white py-2">
        <h6 class="mb-0"><i class="fas fa-history"></i> Historial de cumpleaños atendidos</h6>
    </div>
    <div class="card-body p-3">
        <div class="table-responsive">
            <!-- Botones para exportar PDF o imprimir historial -->
            <div class="d-flex justify-content-end align-items-center gap-2 mb-2 me-2">
                <button id="btnExportPDF" class="btn border border-danger text-danger">

```

```

                <i class="bi bi-file-earmark-pdf"></i> PDF
            </button>
            <button id="btnImprimir" class="btn border border-success text-success">
                <i class="bi bi-printer"></i> Imprimir
            </button>
        </div>

        
        <div id="historialCumplenes" class="p-2"></div>
    </div>
</div>


<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"
></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf-autotable/3.5.29/jspdf.plugin.autotable.min.js"></script>

```

```

|   |— DashboardView.php
<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>CRM Bastos - Dashboard</title>
    <!-- Iconos de Bootstrap -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.5/font/bootstrap-icons.css" />
    <!-- Estilos propios del dashboard -->
    <link rel="stylesheet" href="/CRM_INT/CRM/public/css/Dashboard.css" />
</head>

<!-- Se guarda en data-nombre el usuario de sesión -->
<body data-nombre=<?= $_SESSION['usuario'] ?? 'Usuario' ?>>

<div id="container-dashboard">
    <!-- Aquí se pueden cargar paneles de métricas dinámicas -->
</div>
```

```

<!-- Resumen de métricas en tarjetas -->
<div class="resumen-metricas">

    <!-- Tarjeta de bienvenida al usuario -->
    <div class="tarjeta-metrica">
        <i class="bi bi-person-circle icon-metrica"></i>
        <h4>Bienvenido</h4>
        <p class="valor" id="bienvenida-nombre"></p>
        <span class="subvalor">
            Rol: <span id="bienvenida-rol"></span><br>
            ¡Nos alegra tenerte de vuelta!
        </span>
    </div>

    <!-- Tarjeta de clientes que cumplen años este mes -->
    <div class="tarjeta-metrica">
        <i class="bi bi-gift-fill icon-metrica"></i>
        <h4>Clientes que Cumplen Años este mes</h4>
        <p id="cumple-texto" style="font-size: 0.9em;"></p>
        <a href="CRM_INT/CRM/index.php?view=cumple" class="link-metrica" style="display: block; margin-top: 0.5rem;">
            Ver análisis
        </a>
    </div>

    <!-- Tarjeta del cliente del mes -->
    <div class="tarjeta-metrica">
        <i class="bi bi-award-fill icon-metrica"></i>
        <h4>Cliente del Mes</h4>
        <p class="valor" id="cliente-mes-nombre">N/A</p>
        <span class="subvalor" id="cliente-mes-valor">$0</span>
        <a href="CRM_INT/CRM/index.php?view=analisis&seccion=clientesMayorHistorial" class="link-metrica">
            Ver análisis
        </a>
    </div>

    <!-- Tarjeta de ventas del mes actual -->
    <div class="tarjeta-metrica">
        <i class="bi bi-currency-dollar icon-metrica"></i>
        <h4>Ventas del Mes Actual</h4>
        <p class="valor" id="total-ventas">$0</p>
        <a href="CRM_INT/CRM/index.php?view=analisis&seccion=ventasPorMes" class="link-metrica">
            Ver análisis
        </a>
    </div>

```

```

<!-- Tarjeta del total de clientes -->


<i class="bi bi-people-fill icon-metrica"></i>
    <h4>Total Clientes</h4>
    <p class="valor" id="total-clientes">0</p>



<!-- Tarjeta de ventas del año actual -->


<i class="bi bi-calendar-year icon-metrica"></i>
    <h4>Ventas del Año Actual</h4>
    <p class="valor" id="total-ventas-anio">$0</p>
    <a href="CRM_INT/CRM/index.php?view=analisis&seccion=ventasPorAnio"
    class="link-metrica">
        Ver análisis
    </a>


</div>

<!-- Sección de tareas tipo “to-do list” -->


<h3 class="todo-title">
        <i class="bi bi-pin-angle-fill" style="color: var(--amarillo)"></i>
        Tareas
    </h3>

    <!-- Formulario para agregar una nueva tarea -->
    <form id="formTarea" class="formulario-tarea">
        <input type="text" name="descripcion" id="descripcion"
        placeholder="Escriba una tarea (máx. 220 caracteres)"
        maxlength="220" required />
        <div class="contador-caracteres">
            Caracteres restantes: <span id="contador-caracteres">220</span>
        </div>

        <button type="submit">
            <i class="bi bi-plus-circle-fill"></i> Agregar
        </button>
    </form>

    <!-- Lista de tareas dinámicas (se rellenan con JS) -->
    <ul class="todo-list" id="contenedorTarjetas"></ul>


</div>
</div>

<!-- Scripts necesarios -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

```

```

<!-- Manejo de tareas -->
<script src="/CRM_INT/CRM/public/js/Tarea.js"></script>
<!-- Manejo de métricas y dashboard -->
<script src="/CRM_INT/CRM/public/js/Dashboard.js"></script>
</body>

</html>

```

## | ┌ LayoutView.php

```

<?php
// Lee la vista actual desde la URL (?view=...), si no viene usa
'dashboard'
$vista = $_GET['view'] ?? 'dashboard';
?>
<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <title>CRM Bastos</title>
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <!-- Evita cachear (útil para ver cambios al instante) -->
        <meta http-equiv="Cache-Control" content="no-store" />

        <!-- Bootstrap y Bootstrap Icons (estilos base del sitio) -->
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.6/dist/css/bootstrap.min.
.css" rel="stylesheet">
            <link href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.10.5/font/bootstrap-icons.css" rel="stylesheet">
                <!-- Font Awesome (íconos adicionales) -->
                <link href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css" rel="stylesheet">

        <!-- CSS específico según la vista actual -->
        <?php if ($vista === 'dashboard'): ?>
            <link rel="stylesheet"
href="/CRM_INT/CRM/public/css/Dashboard.css">
        <?php elseif ($vista === 'clientes'): ?>
            <link rel="stylesheet"
href="/CRM_INT/CRM/public/css/Cliente.css">
        <?php elseif ($vista === 'compras'): ?>
            <link rel="stylesheet" href="/CRM_INT/CRM/public/css/Compra.css">
        <?php elseif ($vista === 'codigo'): ?>
            <link rel="stylesheet" href="/CRM_INT/CRM/public/css/Codigo.css">
        <?php elseif ($vista === 'analisis'): ?>

```

```

        <link rel="stylesheet"
href="/CRM_INT/CRM/public/css/Analisis.css">
    <?php elseif ($vista === 'Bitacora'): ?>
        <link rel="stylesheet"
href="/CRM_INT/CRM/public/css/Bitacora.css">
    <?php elseif ($vista === 'cumple'): ?>
        <link rel="stylesheet"
href="/CRM_INT/CRM/public/css/Cumple.css?v=<?= time() ?>">
    <?php elseif ($vista === 'guia'): ?>
        <link rel="stylesheet"
href="/CRM_INT/CRM/public/css/Manual.css?v=<?= time() ?>">
    <?php elseif ($vista === 'guia2'): ?>
        <link rel="stylesheet"
href="/CRM_INT/CRM/public/css/Manual2.css?v=<?= time() ?>">
    <?php endif; ?>

        <!-- CSS general del layout (siempre) -->
        <link rel="stylesheet" href="/CRM_INT/CRM/public/css/Layout.css?v=<?= time() ?>">
            <!-- CSS de usuarios solo si la vista es 'usuarios' (tiene prioridad al ir al final) -->
            <?php if ($vista === 'usuarios'): ?>
                <link rel="stylesheet"
href="/CRM_INT/CRM/public/css/Usuario.css?v=<?= time() ?>">
            <?php endif; ?>
</head>

<body>
    <!-- BOTÓN HAMBURGUESA PARA MÓVIL: abre/cierra el menú lateral -->
    <button class="btn-menu-toggle" id="menuToggle" aria-label="Abrir menú">
        <i class="bi bi-list"></i>
    </button>

    <!-- Sidebar de navegación (menú lateral) -->
    <aside class="sidebar">
        <div class="sidebar-header">
            <!-- Logo del sistema -->
            
            <h3>CRM Bastos</h3>
        </div>

        <!-- Acceso rápido a "Cumpleaños de la semana" -->
        <button type="button" class="btn-cumple-mini" id="btnCumpleMini"
title="Cumpleaños de la semana"
            aria-label="Abrir Cumpleaños">
            <i class="bi bi-gift-fill"></i>

```

```

        </button>

        <!-- Links de navegación; marcan 'active' según $vista -->
        <ul>
            <li>
                <a href="/CRM_INT/CRM/index.php?view=dashboard" class="<?= $vista === 'dashboard' ? 'active' : '' ?>">
                    <i class="bi bi-house"></i> Dashboard
                </a>
            </li>
            <li>
                <a href="/CRM_INT/CRM/index.php?view=clientes" class="<?= $vista === 'clientes' ? 'active' : '' ?>">
                    <i class="bi bi-award-fill"></i> Clientes VIP
                </a>
            </li>
            <li>
                <a href="/CRM_INT/CRM/index.php?view=usuarios" class="<?= $vista === 'usuarios' ? 'active' : '' ?>">
                    <i class="bi bi-person"></i> Usuarios
                </a>
            </li>
            <li>
                <a href="/CRM_INT/CRM/index.php?view=compras" class="<?= $vista === 'compras' ? 'active' : '' ?>">
                    <i class="bi bi-tag"></i> Beneficios
                </a>
            </li>
            <li>
                <a href="/CRM_INT/CRM/index.php?view=codigo" class="<?= $vista === 'codigo' ? 'active' : '' ?>">
                    <i class="bi bi-upc-scan"></i> Código de barras
                </a>
            </li>
            <li>
                <a href="/CRM_INT/CRM/index.php?view=analisis" class="<?= $vista === 'analisis' ? 'active' : '' ?>">
                    <i class="bi bi-graph-up-arrow"></i> Análisis
                </a>
            </li>
            <li>
                <a href="/CRM_INT/CRM/index.php?view=cumple" class="<?= $vista === 'cumple' ? 'active' : '' ?>">
                    <i class="bi bi-gift"></i> Cumpleaños
                    <!-- Badge que indica pendientes (se controla por JS) -->
                    <span id="cumple-badge" style="display:none; margin-left:8px; vertical-align:middle;"></span>
                </a>
            </li>
        </ul>
    
```

```

        </a>
    </li>

    <li>
        <a href="/CRM_INT/CRM/index.php?view=Bitacora" class="<?=
$vista === 'Bitacora' ? 'active' : '' ?>">
            <i class="bi bi-journal-text"></i> Bitácora
        </a>
    </li>
    <li>
        <a href="/CRM_INT/CRM/index.php?view=guia" class="<?=
$vista === 'guia' ? 'active' : '' ?>">
            <i class="bi bi-info-circle"></i> Manual de uso
        </a>
    </li>
    <li>
        <a href="/CRM_INT/CRM/index.php?view=guia2" class="<?=
$vista === 'guia2' ? 'active' : '' ?>">
            <i class="bi bi-person-badge"></i> Manual de usuario
        </a>
    </li>

    <li>
        <!-- Cerrar sesión (controlador en PHP) -->
        <a href="/CRM_INT/CRM/controller/UsuarioController.php?action=logout"
style="color: #dc3545;">
            <i class="bi bi-box-arrow-right"></i> Cerrar Sesión
        </a>
    </li>

```

</ul>

</aside>

<!-- Fondo oscuro para móvil cuando el sidebar está abierto -->

<div class="sidebar-overlay" id="sidebarOverlay"></div>

```

<div class="content">
    <main class="main-content">
        <?php
        // Incluye la vista correspondiente según $vista
        if ($vista === 'clientes') {
            include 'ClienteView.php';
        } elseif ($vista === 'usuarios') {
            include 'UsuarioView.php';
        } elseif ($vista === 'recompensas') {
            include 'RecompensasView.php';
        } elseif ($vista === 'compras') {

```

```

        include 'CompraView.php';
    } elseif ($vista === 'codigo') {
        include 'CodigoView.php';
    } elseif ($vista === 'analisis') {
        include 'AnalisisView.php';
    } elseif ($vista === 'cumple') {
        include 'CumpleView.php';
    } elseif ($vista === 'Bitacora') {
        include 'BitacoraView.php';
    } elseif ($vista === 'guia') {
        include 'ManualView.php';
    } elseif ($vista === 'guia2') {
        include 'Manual2View.php';
    } else {
        // Vista por defecto (dashboard)
        include 'DashboardView.php';
    }
?>

<!-- Modal simple para buscar por número de tarjeta -->
<div id="modalTarjeta" class="modal-tarjeta">
    <div class="modal-contenido">
        <h3>Ingrese número de tarjeta</h3>
        <input type="text" id="modalInputTarjeta"
placeholder="Ej: 60" />
        <div id="modalMensajeError" class="modal-mensaje-
error"></div>
        <div class="modal-botones">
            <button onclick="cerrarModal()">Cancelar</button>
            <button
onclick="redirigirCompra()">Buscar</button>
        </div>
    </div>
</div>

<!-- Modal de "Cumpleaños de la Semana" (se muestra encima
del dashboard) -->
<div id="modalCumples" class="modal-cumples" style="display:
none;">
    <div class="modal-contenido-cumples">
        <div class="modal-header-cumples">
            <h3><i class="bi bi-gift"></i> Cumpleaños de la
Semana</h3>
            <button class="btn-cerrar-cumples"
onclick="cerrarModalCumples()">&times;</button>
        </div>
        <div class="modal-body-cumples">

```

```

        <!-- Aquí se pinta el rango de fechas de la
semana -->
        <div id="rangoCumpleSemana" class="alert alert-
info text-center fw-bold">
            <!-- Aquí se mostrará el rango de la semana --
->
        </div>
        <!-- Aquí se listan los cumpleaños -->
        <div id="listaCumpleanos" class="cumples-
container">
            <!-- Aquí se cargarán los cumpleaños -->
            </div>
        </div>
        <div class="modal-footer-cumples">
            <button class="btn-ir-cumples"
onclick="irACumpleanos()">
                <i class="bi bi-arrow-right-circle"></i> Ir a
Cumpleaños
            </button>
        </div>
    </div>

```

<!-- Scripts Bootstrap (JS de componentes) -->

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.6/dist/js/bootstrap.bundle.min.js"></script>

        <!-- Script general del layout (maneja menú, modales, etc.) --
->
        <script src="/CRM_INT/CRM/public/js/Layout.js"></script>
        <!-- Scanner global (con cache-busting por versión) -->
        <script src="/CRM_INT/CRM/public/js/ScannerGlobal.js?v=<?=
time() ?>"></script>
```

<!-- Scripts específicos por vista (se cargan solo donde aplica) -->

```

<?php if ($vista === 'dashboard'): ?>
<?php elseif ($vista === 'clientes'): ?>
    <script src="/CRM_INT/CRM/public/js/Cliente.js"></script>
<?php endif; ?>

<?php if ($vista === 'cumple'): ?>
    <script src="/CRM_INT/CRM/public/js/Cumple.js"></script>
<?php endif; ?>

<?php if ($vista === 'guia'): ?>
    <script src="/CRM_INT/CRM/public/js/Manual.js"></script>
```

```

<?php endif; ?>

<?php if ($vista === 'guia2'): ?>
    <script src="/CRM_INT/CRM/public/js/Manual2.js"></script>
<?php endif; ?>

<!-- Logout automático al cerrar la pestaña/navegador --&gt;
&lt;script&gt;
    window.addEventListener('beforeunload', function () {
        // Notifica al servidor que se cerró la pestaña
        (cierra sesión)
            navigator.sendBeacon('/CRM_INT/CRM/controller/LogoutOnClose.php');
    });

    // Confirmación manual de cierre de sesión (si se usa)
    function confirmarCerrarSesion() {
        if (confirm("¿Seguro que desea cerrar sesión?")) {
            window.location.href = "index.php?logout=true";
        }
    }
&lt;/script&gt;

&lt;script&gt;
    document.addEventListener('DOMContentLoaded', function ()
{
    // LOGO abre modal de búsqueda por tarjeta (atajo)
    const logo = document.querySelector(".img-header");
    if (logo) {
        logo.style.cursor = 'pointer';
        logo.addEventListener('click', function (e) {
            e.preventDefault();
            if (typeof abrirModal === 'function') {
                abrirModal();          // Abre #modalTarjeta
(definido en Layout.js)
            } else if (typeof mostrarModal ===
'function') {
                mostrarModal();      // Fallback si el
nombre cambia
            }
        });
    }

    // Si estamos en el dashboard: abre automáticamente
el modal de "Cumpleaños"
        // SOLO para el rol 'salonero' (lee el rol desde
localStorage)
        const vistaActual = '&lt;?= $vista ?&gt;';
</pre>

```

```

        if (vistaActual === 'dashboard') {
            let intentos = 0;
            const maxIntentos = 20;
            const intervalo = setInterval(function () {
                const rol =
(localStorage.getItem('rolUsuario') || '').toLowerCase();
                    if (rol === 'salonero') {
                        abrirModalCumple(); // Muestra modal
de cumpleaños al cargar
                        clearInterval(intervalo);
                    }
                    intentos++;
                    if (intentos >= maxIntentos) {
                        clearInterval(intervalo);
                    }
            }, 100);
        }

        // Rol actual guardado en localStorage (minuscula)
        const rol = (localStorage.getItem('rolUsuario') ||
'').toLowerCase();

        // REGLAS DE PERMISOS por rol y vista
        // Admin en Beneficios: no puede eliminar
        if (rol === "administrador" && '<?= $vista ?>' ===
'compras') {
            function disableDeleteButtons() {
                document.querySelectorAll("button").forEach((
btn) => {
                    // Detecta botones de eliminar (por texto
                    o clase)
                    if (btn.textContent.trim().toLowerCase()
=== 'eliminar' || btn.classList.contains('btn-danger')) {
                        btn.disabled = true;
                        btn.classList.add('btn-disabled');
                        btn.title = "No tienes permisos para
eliminar beneficios";
                    }
                });
            }
            // Al cargar
            disableDeleteButtons();
            // Observa cambios en la lista para deshabilitar
nuevos botones
            const lista =
document.getElementById("compraLista") || document.body;
            const observer = new
MutationObserver(disableDeleteButtons);

```

```

        observer.observe(lista, { childList: true,
subtree: true });
    }

        // Salonero en Beneficios: no puede acumular, ni
aplicar descuento, ni eliminar
        if (rol === "salonero" && '<?= $vista ?>' ===
'compras') {

            const observer = new MutationObserver(() => {
                document.querySelectorAll("button").forEach((
btn) => {
                    // Acumular puntos
                    if (btn.textContent.trim().toLowerCase() === 'acumular' || (btn.onclick &&
btn.onclick.toString().includes('acumular'))) {
                        btn.disabled = true;
                        btn.classList.add('btn-disabled');
                        btn.title = "No tienes permisos para
acumular puntos";
                    }
                    // Aplicar descuento
                    if (btn.textContent.trim().toLowerCase() === 'aplicar' || (btn.onclick &&
btn.onclick.toString().includes('descuento'))) {
                        btn.disabled = true;
                        btn.classList.add('btn-disabled');
                        btn.title = "No tienes permisos para
aplicar descuentos";
                    }
                    // Eliminar beneficio
                    if (btn.textContent.trim().toLowerCase() === 'eliminar' || btn.classList.contains('btn-danger')) {
                        btn.disabled = true;
                        btn.classList.add('btn-disabled');
                        btn.title = "No tienes permisos para
eliminar beneficios";
                    }
                });
                // Deshabilita el input de "Acumular compra"
                const inputAcumulada =
document.getElementById("cantidadAcumulada");
                if (inputAcumulada) {
                    inputAcumulada.disabled = true;
                    inputAcumulada.classList.add('form-
readonly');
                    inputAcumulada.title = "No tienes
permisos para acumular compras";
                }
            });
        }
    }
}

```

```

        }
    });

const lista =
document.getElementById("compraLista") || document.body;
observer.observe(lista, { childList: true,
subtree: true });

// Aplicar también al cargar
document.querySelectorAll("button").forEach((btn)
=> {
    if (btn.textContent.trim().toLowerCase() ===
'acumular' || (btn.onclick &&
btn.onclick.toString().includes('acumular'))) {
        btn.disabled = true;
        btn.classList.add('btn-disabled');
        btn.title = "No tienes permisos para
acumular puntos";
    }
    if (btn.textContent.trim().toLowerCase() ===
'aplicar' || (btn.onclick &&
btn.onclick.toString().includes('descuento'))) {
        btn.disabled = true;
        btn.classList.add('btn-disabled');
        btn.title = "No tienes permisos para
aplicar descuentos";
    }
    if (btn.textContent.trim().toLowerCase() ===
'eliminar' || btn.classList.contains('btn-danger')) {
        btn.disabled = true;
        btn.classList.add('btn-disabled');
        btn.title = "No tienes permisos para
eliminar beneficios";
    }
});
// Deshabilitar input al cargar
const inputAcumulada =
document.getElementById("cantidadAcumulada");
if (inputAcumulada) {
    if (inputAcumulada) {
        inputAcumulada.disabled = true;
        inputAcumulada.classList.add('form-
readonly');
        inputAcumulada.title = "No tienes permisos
para acumular compras";
    }
}

if (rol === "salonero") {
    // Enlace Usuarios: sin acceso
}

```

```

        const linkUsuarios =
document.getElementById('link-usuarios');
        if (linkUsuarios) {
            linkUsuarios.classList.add('disabled-link');
            linkUsuarios.removeAttribute('href');
            linkUsuarios.title = "Acceso restringido";
        }
        // Enlace Análisis: sin acceso
        const linkAnalisis =
document.getElementById('link-analisis');
        if (linkAnalisis) {
            linkAnalisis.classList.add('disabled-link');
            linkAnalisis.removeAttribute('href');
            linkAnalisis.title = "Acceso restringido";
        }
        // Enlace Bitácora: sin acceso
        const linkBitacora =
document.getElementById('link-Bitacora');
        if (linkBitacora) {
            linkBitacora.classList.add('disabled-link');
            linkBitacora.removeAttribute('href');
            linkBitacora.title = "Acceso restringido";
        }

        // Vista actual
const currentView = '<?= $vista ?>';

        // En dashboard: oculta las tarjetas de métricas
para salonero
        if (currentView === 'dashboard') {
            document.querySelectorAll('.tarjeta-
metrica').forEach(div => {
                div.style.display = 'none';
            });
        }

        // En clientes: todo en solo lectura para
salonero
        if (currentView === 'clientes') {
            const submitBtn =
document.getElementById('submitBtn');
            if (submitBtn) {
                submitBtn.disabled = true;
                submitBtn.classList.add('btn-disabled');
                submitBtn.title = "No tienes permisos
para guardar clientes";
            }
        }
    
```

```

        const cancelBtn =
document.getElementById('cancelBtn');
        if (cancelBtn) {
            cancelBtn.disabled = true;
            cancelBtn.classList.add('btn-disabled');
            cancelBtn.title = "No tienes permisos
para editar clientes";
        }

        const form =
document.getElementById('clienteForm');
        if (form) {
            const campos =
form.querySelectorAll("input, select, textarea");
            campos.forEach((campo) => {
                campo.readOnly = true;
                campo.disabled = true;
                campo.classList.add('form_READONLY');
            });
        }

        // Observa y deshabilita botones de
editar/eliminar/reasignar que aparezcan
        const observer = new MutationObserver(() => {
            document.querySelectorAll("button.btn-
warning").forEach((btn) => {
                btn.disabled = true;
                btn.classList.add('btn-disabled');
                btn.title = "No tienes permisos para
editar clientes";
            });
            document.querySelectorAll("button.btn-
danger").forEach((btn) => {
                btn.disabled = true;
                btn.classList.add('btn-disabled');
                btn.title = "No tienes permisos para
eliminar clientes";
            });
        });
        // Reasignar
        document.querySelectorAll('button').forEa
ch((btn) => {
        if (btn.textContent.trim() ===
'Reasignar' || (btn.onclick &&
btn.onclick.toString().includes('abrirModalReasignar')))) {
            btn.disabled = true;
            btn.classList.add('btn-
disabled');

```

```

        btn.title = "No tienes permisos
para reasignar clientes";
    }
}
});

const lista =
document.getElementById("clienteLista");
if (lista) {
    observer.observe(lista, { childList:
true, subtree: true });
}
}

// En código de barras: deshabilita imprimir para
salonero
if (currentView === 'codigo') {
    function disableImprimirBtns() {
        document.querySelectorAll('button').forEach((btn) => {
            if (btn.onclick &&
btn.onclick.toString().includes('imprimirCodigo')) {
                btn.disabled = true;
                btn.classList.add('btn-
disabled');
                btn.title = "No tienes permisos
para imprimir";
            }
        });
    }
    // Expone la función (por si se llama desde
otros lados)
window.disableImprimirBtns =
disableImprimirBtns;
    // Observa la lista e inhabilita nuevos
botones
    const observer = new
MutationObserver(disableImprimirBtns);
    const lista =
document.getElementById("codigoLista");
    if (lista) {
        observer.observe(lista, { childList:
true, subtree: true });
    }
    // Ejecuta al cargar
    disableImprimirBtns();
}
} else if (rol === "administrador") {
}

```

```

        // Admin: restringe Usuarios y Bitácora (sin
acceso directo desde menú)
        const linkUsuarios =
document.getElementById('link-usuarios');
        if (linkUsuarios) {
            linkUsuarios.classList.add('disabled-link');
            linkUsuarios.removeAttribute('href');
            linkUsuarios.title = "Acceso restringido";
        }

        const linkBitacora =
document.getElementById('link-Bitacora');
        if (linkBitacora) {
            linkBitacora.classList.add('disabled-link');
            linkBitacora.removeAttribute('href');
            linkBitacora.title = "Acceso restringido";
        }

        const currentView = '<?= $vista ?>';

        // Admin en clientes: no puede eliminar ni
reasignar
        if (currentView === 'clientes') {
            const observer = new MutationObserver(() => {
                document.querySelectorAll("button.btn-
danger").forEach((btn) => {
                    btn.disabled = true;
                    btn.classList.add('btn-disabled');
                    btn.title = "No tienes permisos para
eliminar clientes";
                });
            });

            document.querySelectorAll('button').forEa
ch((btn) => {
                if (btn.textContent.trim() ===
'Reasignar' || (btn.onclick &&
btn.onclick.toString().includes('abrirModalReasignar')))) {
                    btn.disabled = true;
                    btn.classList.add('btn-
disabled');
                    btn.title = "No tienes permisos
para reasignar clientes";
                }
            });
        });
    });

```

```

        const lista =
document.getElementById("clienteLista");
        if (lista) {
            observer.observe(lista, { childList:
true, subtree: true });
        }
    }

    // Admin en código de barras: no puede imprimir
if (currentView === 'codigo') {
    function disableImprimirBtns() {
        document.querySelectorAll('button').forEach((btn) => {
            if (btn.onclick &&
btn.onclick.toString().includes('imprimirCodigo')) {
                btn.disabled = true;
                btn.classList.add('btn-
disabled');
                btn.title = "No tienes permisos
para imprimir";
            }
        });
    }
    window.disableImprimirBtns =
disableImprimirBtns;
    const lista =
document.getElementById("codigoLista");
    if (lista) {
        const observer = new
MutationObserver(disableImprimirBtns);
        observer.observe(lista, { childList:
true, subtree: true });
    }

    disableImprimirBtns();
}
} else if (rol === "propietario") {
    // Propietario: acceso total (sin restricciones)
}

        // Código antiguo comentado (duplicado/obsoleto). Se
deja como referencia.
        // const rolUsuario = "<?= isset($_SESSION['rol']) ?"
strtolower($_SESSION['rol']) : '' ?>";
        // const vistaActual = "<?= $vista ?>";
        // if (rolUsuario === "salonero" && vistaActual ===
"dashboard") {
            //     abrirModal(); // Esta función ya no existe

```

```

        // }
        // const logo = document.querySelector(".img-
header");
        // if (logo) {
        //     logo.style.cursor = 'pointer';
        //     logo.addEventListener('click', function () {
        //         abrirModal(); // Esta función ya no existe
        //     });
        // }
    });
</script>

<script>
    // Muestra/oculta el "punto" de pendientes en el menú
Cumpleaños
    function mostrarCumpleBadge(pendientes) {
        const badge = document.getElementById('cumple-
badge');
        if (!badge) return;
        if (pendientes) {
            badge.style.display = 'inline-block';
            badge.innerHTML = `<span style="display:inline-
block;width:12px;height:12px;background:#f9c41f;border-
radius:50%;border:2px solid #000;box-shadow:0 0 2px #000;vertical-
align:middle;"></span>`;
        } else {
            badge.style.display = 'none';
            badge.innerHTML = '';
        }
    }
    document.addEventListener('DOMContentLoaded', function ()
{
    // Consulta al backend los cumpleaños de la semana y
activa el badge si hay "Activos"
    fetch('/CRM_INT/CRM/controller/CumpleController.php',
{
    method: 'POST',
    headers: { 'Content-Type': 'application/x-www-
form-urlencoded' },
    body: 'action=readSemana'
})
    .then(res => res.json())
    .then(data => {
        if (data.success && Array.isArray(data.data)
&& data.data.some(c => c.estado === 'Activo')) {
            mostrarCumpleBadge(true);
        } else {
            mostrarCumpleBadge(false);
        }
    });
}
);

```

```

        }
    })
    .catch(() => mostrarCumpleBadge(false));
});

</script>
<script>
    document.addEventListener('DOMContentLoaded', function ()
{
    // Botón mini de Cumpleaños (en header): abre el
    modal de cumple
    const btnCumpleMini =
document.getElementById('btnCumpleMini');
    if (btnCumpleMini && typeof abrirModalCumple ===
'function') {
        btnCumpleMini.addEventListener('click', function
(e) {
            e.preventDefault();
            abrirModalCumple(); // Abre el modal con
id="modalCumple"
        });
    }
});
</script>

</body>
</html>
```

```

|   |— LoginView.php
<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Portal de Acceso - Sistema de Gestión</title>
    <!-- Estilos del login -->
    <link rel="stylesheet" href="public/css/Login.css">
</head>

<body>
    <!-- Contenedor principal del login -->
    <div class="login-container">
        <!-- Tarjeta del formulario -->
        <div class="login-card">
```

```

        <!-- Logo superior -->
        <div class="logo-container">
            
        </div>

        <!-- Encabezado: título y subtítulo -->
        <div class="header">
            <h2>CRM BASTOS</h2>
            <div class="subtitle">Inicio de sesión</div>
        </div>

        <!-- Formulario de acceso (el envío/validación lo maneja
Login.js) -->
        <form id="loginForm" class="login-form">
            <!-- Usuario -->
            <div class="form-group">
                <label for="email">Nombre de Usuario</label>
                <input type="text" id="email" name="usuario"
placeholder="Ej. CarlosBastos" required>
                <!-- Mensaje de error para el usuario -->
                <span class="error-message" id="emailError"></span>
            </div>

            <!-- Contraseña -->
            <div class="form-group">
                <label for="password">Contraseña</label>
                <input type="password" id="password"
name="contrasena" placeholder="Contraseña" required>
                <!-- Mensaje de error para la contraseña -->
                <span class="error-message"
id="passwordError"></span>
            </div>

            <!-- Botón de enviar con spinner de carga -->
            <button type="submit" class="btn-primary" id="loginBtn">
                <span class="btn-text">Iniciar Sesión</span>
                <div class="spinner" id="spinner"></div>
            </button>
        </form>
    </div>
</div>

        <!-- Lógica de login: validación y envío -->
        <script src="public/js/Login.js"></script>
</body>

</html>

```

## | └─ ManualView.php

```
<div class="container guia-container mt-4">
    <h2 class="guia-titulo text-center mb-4">
        <i class="bi bi-info-circle"></i> Manual de uso del sistema
    </h2>

    <div class="row row-cols-1 row-cols-md-2 g-4">
        <!-- DASHBOARD -->
        <div class="col">
            <div class="guia-card shadow">
                <div class="guia-header">
                    <i class="bi bi-house"></i>
                    <span class="guia-link-seccion" data-ir-
a="?view=dashboard">Dashboard</span>
                </div>
                <div class="guia-body">
                    Aquí ves datos rápidos como ventas y cumpleaños.
                    Propietarios y admins ven métricas exclusivas.
                    También puedes gestionar tareas pendientes.
                    Saloneros solo acceden a sus tareas del día.
                </div>
                <!-- Video con lazy loading -->
                <div class="guia-video mt-2">
                    <div class="video-placeholder" data-video-id="CIjYfjxSJbI">
                        
                        <div class="play-button">
                            <i class="bi bi-play-circle-fill"></i>
                        </div>
                    </div>
                </div>
            </div>
        </div>

        <!-- CLIENTES VIP -->
        <div class="col">
            <div class="guia-card shadow">
                <div class="guia-header">
                    <i class="bi bi-award-fill"></i>
                    <span class="guia-link-seccion" data-ir-
a="?view=clientes">Clientes VIP</span>
                </div>
                <div class="guia-body">
                    Puedes registrar, editar y reasignar tarjetas.
                    Admins solo pueden ver y editar información.
                </div>
            </div>
        </div>
    </div>
</div>
```

```

    Saloneros pueden ver, pero no modificar nada.
    Las tarjetas reasignadas cambian su código.
</div>
<<<<< Updated upstream
    <!-- Video con lazy loading -->
    <div class="guia-video mt-2">
        <div class="video-placeholder" data-video-id="ozohxbb4A5A">
            
            <div class="play-button">
                <i class="bi bi-play-circle-fill"></i>
            </div>
        </div>
    </div>
=====
    <!-- Video explicativo -->
    <div class="guia-video mt-2">
        <iframe width="100%" height="200"
src="https://www.youtube.com/embed/DjKJVifdwe8?si=o88zWy0QIt8taato"
            title="Video Dashboard" frameborder="0"
allowfullscreen></iframe>
        </div>
    >>>>> Stashed changes
        </div>
    </div>

    <!-- USUARIOS -->
    <div class="col">
        <div class="guia-card shadow">
            <div class="guia-header">
                <i class="bi bi-person"></i>
                <span class="guia-link-seccion" data-ir-
a="?view=usuarios">Usuarios</span>
            </div>
            <div class="guia-body">
                Solo el propietario accede a esta sección.
                Aquí se crean usuarios con roles y claves.
                Puedes editar, eliminar y buscar usuarios.
                Los usuarios con el rol salonero o administrador,
                no tendrán acceso.
            </div>
            <!-- Video con lazy loading -->
            <div class="guia-video mt-2">
                <div class="video-placeholder" data-video-id="p7yA0D90Qj0">
                    

```

```

        <div class="play-button">
            <i class="bi bi-play-circle-fill"></i>
        </div>
    </div>
</div>

<!-- BENEFICIOS -->
<div class="col">
    <div class="guia-card shadow">
        <div class="guia-header">
            <i class="bi bi-tag"></i>
            <span class="guia-link-seccion" data-ir-a="?view=compras">Beneficios</span>
        </div>
        <div class="guia-body">
            Busca un cliente para registrar su compra.
            Puedes acumular puntos o aplicar descuentos.
            El historial se guarda debajo automáticamente.
            Solo el propietario puede eliminar registros.
        </div>
        <!-- Video con lazy loading -->
        <div class="guia-video mt-2">
            <div class="video-placeholder" data-video-id="5cDW7KrH2iI">
                
                <div class="play-button">
                    <i class="bi bi-play-circle-fill"></i>
                </div>
            </div>
        </div>
    </div>
</div>

<!-- CÓDIGO DE BARRAS -->
<div class="col">
    <div class="guia-card shadow">
        <div class="guia-header">
            <i class="bi bi-upc-scan"></i>
            <span class="guia-link-seccion" data-ir-a="?view=codigo">Código de barras</span>
        </div>
        <div class="guia-body">
            Imprime o guarda el código de cada tarjeta.
            Escanearlo abre directamente el perfil cliente.
            El sistema detecta la tarjeta automáticamente.
        </div>
    </div>
</div>

```

```

        Ahorra tiempo y evita errores en la búsqueda.

    </div>
    <!-- Video con lazy loading -->
    <div class="guia-video mt-2">
        <div class="video-placeholder" data-video-id="7QINQ76tyZA">
            
            <div class="play-button">
                <i class="bi bi-play-circle-fill"></i>
            </div>
        </div>
    </div>
</div>

<!-- ANÁLISIS -->
<div class="col">
    <div class="guia-card shadow">
        <div class="guia-header">
            <i class="bi bi-graph-up-arrow"></i>
            <span class="guia-link-seccion" data-ir-
a="?view=analisis">Análisis</span>
        </div>
        <div class="guia-body">
            Solo propietarios y admins pueden ingresar.
            Muestra gráficos de clientes y estadísticas.
            Puedes buscar por nombre o zona de origen.
            Incluye ventas por mes, año y más métricas.
        </div>
        <!-- Video con lazy loading -->
        <div class="guia-video mt-2">
            <div class="video-placeholder" data-video-id="OmVzLpIc1Gc">
                
                <div class="play-button">
                    <i class="bi bi-play-circle-fill"></i>
                </div>
            </div>
        </div>
    </div>
</div>

<!-- CUMPLEAÑOS -->
<div class="col">
    <div class="guia-card shadow">
        <div class="guia-header">

```

```

        <i class="bi bi-gift"></i>
        <span class="guia-link-seccion" data-ir-
a="?view=cumple">Cumpleaños</span>
    </div>
    <div class="guia-body">
        Muestra clientes que cumplen años esta semana.
        Puedes enviar correos o registrar llamadas.
        El estado cambia a "lista" si ya fue atendido.
        Todos los roles pueden acceder sin restricción.
    </div>
<<<<< Updated upstream
        <!-- Video con lazy loading -->
        <div class="guia-video mt-2">
            <div class="video-placeholder" data-video-id="Bcup9Lkpank">
                
                <div class="play-button">
                    <i class="bi bi-play-circle-fill"></i>
                </div>
            </div>
        </div>
=====

        <!-- Video explicativo -->
        <div class="guia-video mt-2">
            <iframe width="100%" height="200"
src="https://www.youtube.com/embed/RdXPauEavEI?si=AISeq904B6eXFK-C"
                title="Video Dashboard" frameborder="0"
allowfullscreen></iframe>
        </div>
>>>>> Stashed changes
        </div>
    </div>

        <!-- BITÁCORA -->
        <div class="col">
            <div class="guia-card shadow">
                <div class="guia-header">
                    <i class="bi bi-journal-text"></i>
                    <span class="guia-link-seccion" data-ir-
a="?view=Bitacora">Bitácora</span>
                </div>
                <div class="guia-body">
                    Solo el propietario puede ver esta sección.
                    Muestra quién entró, cuándo y cuánto duró.
                    Puedes descargar en Word o en PDF si deseas.
                    Los datos se eliminan automáticamente al mes.
                </div>

```

```

<<<<< Updated upstream
    <!-- Video con lazy loading -->
    <div class="guia-video mt-2">
        <div class="video-placeholder" data-video-id="pkPX4K3yCd8">
            
            <div class="play-button">
                <i class="bi bi-play-circle-fill"></i>
            </div>
        </div>
    </div>
=====
        <!-- Video explicativo -->
    <div class="guia-video mt-2">
        <iframe width="100%" height="200"
src="https://www.youtube.com/embed/zqhE6sXgofM?si=oIbr8m_06YQaiXsI"
            title="Video Dashboard" frameborder="0"
allowfullscreen></iframe>
    </div>
>>>>> Stashed changes
    </div>
</div>

        <!-- LOGO DE BASTOS -->
<div class="col">
    <div class="guia-card shadow">
        <div class="guia-header">
            <i class="bi bi-box-arrow-in-down-left"></i>
            <span class="guia-link-seccion" onclick="abrirModal()">Logo de
Bastos</span>
        </div>
        <div class="guia-body">
            Al hacer clic en el logo se abre una ventana.
            Puede ver quien cumpleaños.
            El sistema te lleva directo a la sección de cumpleaños.
        </div>
    </div>
</div>
</div>

<!-- CERRAR SESIÓN -->
<div class="col">
    <div class="guia-card shadow">
        <div class="guia-header">
            <i class="bi bi-box-arrow-right"></i>
            <span class="guia-link-seccion"
onclick="window.location.href='/CRM_INT/CRM/controller/UsuarioController.
php?action=logout'">

```

```

        Cerrar Sesión
    </span>
</div>
<div class="guia-body">
    Este botón cierra tu sesión de forma segura.
    Si estás inactivo por 30 minutos, se cerrará.
    Para volver a entrar, pide tu contraseña.
    Ayuda a proteger la información del sistema.
</div>
</div>
</div>

<style>
.video-placeholder {
    position: relative;
    cursor: pointer;
    transition: transform 0.2s ease;
}

.video-placeholder:hover {
    transform: scale(1.02);
}

.play-button {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    color: #ff0000;
    font-size: 3rem;
    text-shadow: 0 0 10px rgba(0,0,0,0.7);
    transition: all 0.3s ease;
}

.video-placeholder:hover .play-button {
    transform: translate(-50%, -50%) scale(1.1);
}

.guia-video iframe {
    border-radius: 8px;
}
</style>

```

| └─ Manual2View.php

```
<div class="container guia-container mt-4">
    <h2 class="guia-titulo text-center mb-4">
```

```

<i class="bi bi-info-circle"></i> Manual de usuario para el sistema
</h2>

<!-- Acordeones del manual -->
<div class="guia-accordion">

    <!-- 1) Requerimientos (cerrado por defecto) -->
    <details class="acc-item">
        <summary class="acc-summary">
            <span><i class="bi bi-gear"></i> Requerimientos para el
levantamiento y uso del sistema</span>
            <i class="bi bi-chevron-down acc-arrow"></i>
        </summary>

        <div class="acc-content">
            <ul class="mb-3">
                <li>Tener <strong>XAMPP</strong> instalado con los servicios
<strong>Apache</strong> y <strong>MySQL</strong> activos.</li>
                <li>Contar con un cliente/administrador de base de datos:
<em>HeidiSQL</em>, <em>phpMyAdmin</em>, <em>SQLyog</em> o <em>MySQL
Workbench</em>. </li>
                <li>Tener la base de datos del sistema debidamente importada en
el motor de MySQL.</li>
                <li>Una computadora con navegador web actualizado
(Windows/macOS/Linux) para acceder al sistema.</li>
            </ul>
        </div>
    </details>

    <!-- 2) Requerimientos de hardware (nuevo, cerrado por defecto) -->
    <details class="acc-item">
        <summary class="acc-summary">
            <span><i class="bi bi-cpu"></i> Requerimientos de hardware</span>
            <i class="bi bi-chevron-down acc-arrow"></i>
        </summary>

        <div class="acc-content">
            <ul class="mb-3">
                <li>
                    <strong>Lector de código de barras:</strong>
                    Recomendaciones: <em>Eyoyo EY-007</em> (aprox. ₩25,000) o
<em>Inateck BCST-70</em> (aprox. ₩35,000).
                </li>
                <li>
                    <strong>Impresora de código de barras:</strong>
                    Se recomienda <em>Xprinter XP-350B</em> por su relación
calidad/precio (aprox. ₩65,000 - ₩80,000).<br>
                </li>
            </ul>
        </div>
    </details>

```

```

        <strong>Consumible:</strong> Rollo de etiquetas <em>40 × 30
mm</em> (aprox. ₡6,000 - ₡8,000).
    </li>
    <li>
        <strong>Laptop (requerimientos mínimos):</strong>
        <ul>
            <li><strong>Procesador:</strong> Intel® Core™ i5-
1145G7</li>
            <li><strong>Frecuencia base / turbo:</strong> 2.60 GHz / hasta ~4.40 GHz</li>
            <li><strong>Núcleos / hilos:</strong> 4 / 8</li>
            <li><strong>Gráficos integrados:</strong> Intel Iris Xe Graphics</li>
            <li><strong>Memoria RAM:</strong> 16 GB (recomendado 24 GB)</li>
            <li><strong>Almacenamiento:</strong> 256 GB SSD</li>
        </ul>
    </li>
</ul>
<p class="nota">Precios referenciales en colones costarricenses (₡). Ajustar según proveedor/disponibilidad.</p>
</div>
</details>

<!-- 3) Puesta en funcionamiento (cerrado por defecto) -->
<details class="acc-item">
    <summary class="acc-summary">
        <span><i class="bi bi-list-check"></i> Puesta en funcionamiento del sistema</span>
        <i class="bi bi-chevron-down acc-arrow"></i>
    </summary>

    <div class="acc-content">
        <ol class="pasos-list">
            <li>Verificar conexión a internet.</li>
            <li>Abrir el navegador de preferencia.</li>
            <li>Descargar <strong>XAMPP</strong>:  

https://www.apachefriends.org/download.html</li>
                <li>Click en la primera versión <strong>Esperar a que se descargue</strong></li>
                <li>Buscar en descargas</li>
                <li>Instalar XAMPP y activar <strong>Apache</strong> y <strong>MySQL</strong></li>
                <li>Descargar <strong>HeidiSQL</strong> en:  

https://www.heidisql.com/download.php</li>
                    <li><strong>Guíarse con el video para la descarga de HeidiSQL</strong></li>

```

```

<li>Instalar XAMPP y activar <strong>Apache</strong> y
<strong>MySQL</strong></li>
    <li>Descargar la base de datos (.sql): <strong>Ver video de
cómo importar la bd</strong></li>
        <li>Importar la base de datos en <em>HeidiSQL</em> o
<em>phpMyAdmin</em>.</li>
            <li>Copiar el proyecto a <code>htdocs</code> (carpeta de
XAMPP).</li>
                <li>Abrir la ruta del proyecto:
<code>http://localhost/CRM_INT/CRM/</code></li>
                    <li>Configurar credenciales en <code>config/Database.php</code>
para establecer una contraseña.</li>
                <li>Probar inicio de sesión y navegación básica.</li>
            </ol>

        <!-- Videos (ocultos por defecto) -->
        <details class="acc-item nested">
            <summary class="acc-summary">
                <span><i class="bi bi-play-circle"></i> Videos (instalación y
primeros pasos)</span>
                <i class="bi bi-chevron-down acc-arrow"></i>
            </summary>
            <div class="acc-content">

                <!-- Video 1: XAMPP -->
                <div class="guia-video mt-2">
<<<<<< Updated upstream
                <div class="video-placeholder-generic" data-titulo="Cómo
descargar e instalar XAMPP">
                    <div class="video-preview">
                        <div class="video-icon">
                            <i class="bi bi-download"></i>
                        </div>
                        <div class="video-info">
                            <h6>Cómo descargar e instalar XAMPP</h6>
                            <p>Guía paso a paso para la instalación</p>
                        </div>
                        <div class="play-button-generic">
                            <i class="bi bi-play-circle-fill"></i>
                        </div>
                    </div>
                </div>
                =====
                <iframe width="100%" height="200"
src="https://www.youtube.com/embed/03msE0cHP2E?si=ituUDK3
jzRZRbdRG"
                    title="Cómo descargar e instalar XAMPP" frameborder="0"
allowfullscreen></iframe>

```

```

>>>>> Stashed changes
    </div>

        <!-- Video 2: Apache/MySQL -->
        <div class="guia-video mt-2">
<<<<< Updated upstream
            <div class="video-placeholder-generic" data-titulo="Cómo
activar servicios Apache/MySQL">
                <div class="video-preview">
                    <div class="video-icon">
                        <i class="bi bi-server"></i>
                    </div>
                    <div class="video-info">
                        <h6>Cómo activar servicios Apache/MySQL</h6>
                        <p>Configuración de servicios básicos</p>
                    </div>
                    <div class="play-button-generic">
                        <i class="bi bi-play-circle-fill"></i>
                    </div>
                </div>
            </div>
=====
            <iframe width="100%" height="200"
src="https://www.youtube.com/embed/p1uZlYBueX0?si=385Qw_S
HDxGXNhq"
title="Cómo activar servicios Apache/MySQL -"
frameborder="0" allowfullscreen></iframe>
>>>>> Stashed changes
    </div>

        <!-- Video 3: Credenciales BD -->
        <div class="guia-video mt-2">
<<<<< Updated upstream
            <div class="video-placeholder-generic" data-titulo="Cómo
poner credenciales a la base de datos">
                <div class="video-preview">
                    <div class="video-icon">
                        <i class="bi bi-key"></i>
                    </div>
                    <div class="video-info">
                        <h6>Cómo poner credenciales a la base de datos</h6>
                        <p>Configuración de seguridad</p>
                    </div>
                    <div class="play-button-generic">
                        <i class="bi bi-play-circle-fill"></i>
                    </div>
                </div>
            </div>
        
```

```

=====
<iframe width="100%" height="200"
        src="https://www.youtube.com/embed//jHxqhRftB1U?si=h0nm5qv5L6Huz728"
        title="Cómo Poner credenciales a la base de datos"
frameborder="0" allowfullscreen></iframe>
>>>>> Stashed changes
</div>

        <!-- Video 4: Importar BD -->
        <div class="guia-video mt-2">
<<<<< Updated upstream
        <div class="video-placeholder-generic" data-titulo="Cómo
importar la base de datos">
            <div class="video-preview">
                <div class="video-icon">
                    <i class="bi bi-database"></i>
                </div>
                <div class="video-info">
                    <h6>Cómo importar la base de datos</h6>
                    <p>Proceso de importación</p>
                </div>
                <div class="play-button-generic">
                    <i class="bi bi-play-circle-fill"></i>
                </div>
            </div>
        </div>
=====

        <iframe width="100%" height="200"
        src="https://www.youtube.com/embed/M9FpsTmXZkw?si=vMVkfJQ6JygbZVAR"
        title="Cómo importar la base de datos" frameborder="0"
allowfullscreen></iframe>
>>>>> Stashed changes
</div>

        <!-- Video 5: Ruta del proyecto -->
        <div class="guia-video mt-2">
<<<<< Updated upstream
        <div class="video-placeholder-generic" data-titulo="Cómo
entrar a la ruta del proyecto">
            <div class="video-preview">
                <div class="video-icon">
                    <i class="bi bi-folder"></i>
                </div>
                <div class="video-info">
                    <h6>Cómo entrar a la ruta del proyecto</h6>
                    <p>Acceso al sistema local</p>

```

```

        </div>
        <div class="play-button-generic">
            <i class="bi bi-play-circle-fill"></i>
        </div>
    </div>
=====

<iframe width="100%" height="200"
        src="https://www.youtube.com/embed/jHxqhRftB1U?si=tGzvPcH
Cr18Vi5mb"
        title="Cómo Entrar a la ruta del proyecto"
frameborder="0" allowfullscreen></iframe>
>>>>> Stashed changes
        </div>

        <!-- Video 6: Respaldos -->
        <div class="guia-video mt-2">
<<<<< Updated upstream
        <div class="video-placeholder-generic" data-titulo="Cómo
hacer respaldos de la base de datos">
            <div class="video-preview">
                <div class="video-icon">
                    <i class="bi bi-shield-check"></i>
                </div>
                <div class="video-info">
                    <h6>Cómo hacer respaldos de la base de datos</h6>
                    <p>Backup y restauración</p>
                </div>
                <div class="play-button-generic">
                    <i class="bi bi-play-circle-fill"></i>
                </div>
            </div>
        </div>
=====

<iframe width="100%" height="200"
        src="https://www.youtube.com/embed/LcgjCTqtwnE?si=gSXaHrf
MfCiTIo34"
        title="Cómo hacer respaldos de la base de datos"
frameborder="0" allowfullscreen></iframe>
>>>>> Stashed changes
        </div>

        <!-- Video 7: Acceso con teléfono (config) -->
        <div class="guia-video mt-2">
<<<<< Updated upstream
        <div class="video-placeholder-generic" data-titulo="Cómo
configurar el acceso para entrar con teléfono al sistema">
            <div class="video-preview">

```

```

<div class="video-icon">
    <i class="bi bi-phone"></i>
</div>
<div class="video-info">
    <h6>Configurar acceso con teléfono</h6>
    <p>Configuración de acceso móvil</p>
</div>
<div class="play-button-generic">
    <i class="bi bi-play-circle-fill"></i>
</div>
</div>
=====

<iframe width="100%" height="200"
src="https://www.youtube.com/embed/eCUP5z4fyM4?si=Z_qNy9y
9Fq8sfrLm"
title="Cómo configurar el acceso para entrar con teléfono
al sistema" frameborder="0" allowfullscreen></iframe>
>>>>> Stashed changes
</div>

<!-- Video 8: Acceso con teléfono --&gt;
&lt;div class="guia-video mt-2"&gt;
&lt;&lt;&lt;&lt;&lt; Updated upstream
    &lt;div class="video-placeholder-generic" data-titulo="Cómo
entrar al sistema con el teléfono"&gt;
        &lt;div class="video-preview"&gt;
            &lt;div class="video-icon"&gt;
                &lt;i class="bi bi-phone-fill"&gt;&lt;/i&gt;
            &lt;/div&gt;
            &lt;div class="video-info"&gt;
                &lt;h6&gt;Cómo entrar al sistema con el teléfono&lt;/h6&gt;
                &lt;p&gt;Acceso móvil al sistema&lt;/p&gt;
            &lt;/div&gt;
            &lt;div class="play-button-generic"&gt;
                &lt;i class="bi bi-play-circle-fill"&gt;&lt;/i&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;
=====

&lt;iframe width="100%" height="200"
src="https://www.youtube.com/embed/kmNUa1KKs88?si=7KqX5-
3Ra_Kd0HU5"
title="Cómo entrar al sistema con el teléfono"
frameborder="0" allowfullscreen&gt;&lt;/iframe&gt;
&gt;&gt;&gt;&gt;&gt; Stashed changes
&lt;/div&gt;
</pre>

```

```

        </div>
    </details>
</div>
</details>

</div>
</div>

<style>
/* Estilos para videos placeholders genéricos */
.video-placeholder-generic {
    cursor: pointer;
    transition: all 0.3s ease;
    border-radius: 8px;
    overflow: hidden;
}

.video-preview {
    display: flex;
    align-items: center;
    padding: 15px;
    background: linear-gradient(135deg, #f8f9fa 0%, #e9ecef 100%);
    border: 2px solid #dee2e6;
    border-radius: 8px;
    min-height: 80px;
    position: relative;
    transition: all 0.3s ease;
}

.video-placeholder-generic:hover .video-preview {
    background: linear-gradient(135deg, #e3f2fd 0%, #bbdefb 100%);
    border-color: #2196f3;
    transform: translateY(-2px);
    box-shadow: 0 4px 12px rgba(33, 150, 243, 0.3);
}

.video-icon {
    font-size: 2rem;
    color: #6c757d;
    margin-right: 15px;
    transition: color 0.3s ease;
}

.video-placeholder-generic:hover .video-icon {
    color: #2196f3;
}

.video-info {

```

```
    flex: 1;
}

.video-info h6 {
  margin: 0 0 5px 0;
  color: #495057;
  font-weight: 600;
}

.video-info p {
  margin: 0;
  color: #6c757d;
  font-size: 0.9rem;
}

.play-button-generic {
  font-size: 2.5rem;
  color: #ff0000;
  transition: all 0.3s ease;
}

.video-placeholder-generic:hover .play-button-generic {
  transform: scale(1.1);
  color: #d32f2f;
}

.guia-video iframe {
  border-radius: 8px;
}

/* Mensaje cuando no hay video real */
.video-message {
  background: #fff3cd;
  border: 1px solid #ffea7;
  border-radius: 8px;
  padding: 20px;
  text-align: center;
  color: #856404;
}

.video-message i {
  font-size: 3rem;
  margin-bottom: 10px;
  display: block;
}
</style>

<script>
```

```

document.addEventListener("DOMContentLoaded", function () {
    // Lazy loading para videos genéricos (sin URL real)
    document.querySelectorAll(".video-placeholder-generic").forEach((placeholder) => {
        placeholder.addEventListener("click", function() {
            const titulo = this.getAttribute("data-titulo");

            // Crear mensaje explicativo ya que no hay URLs reales
            const videoMessage = document.createElement("div");
            videoMessage.className = "video-message";
            videoMessage.innerHTML =
                `

###### ${titulo}



Este video estará disponible próximamente.<br>
                Cuando tengas las URLs reales de YouTube, podrás
                reemplazar este placeholder.</small>


`;

            // Reemplazar el placeholder con el mensaje
            this.parentNode.replaceChild(videoMessage, this);
        });
    });
});
</script>

```

## | └ UsuarioView.php

```

<!-- Líneas decorativas -->
<div class="decor-line" style="top: 120px; left: 40%;"></div>
<div class="decor-line" style="top: 250px; left: 20%; width:
80px;"></div>

<div class="container mt-4 mover-derecha">

    <!-- Formulario de registro/edición de usuario -->
    <div class="card shadow mb-4">
        <div class="card-header usuarios-header d-flex align-items-
center">
            <h5 class="mb-0 flex-grow-1 titulo-usuarios">Registro de
Usuario</h5>
        </div>
        <div class="card-body">
            <form id="usuarioForm">
                <!-- ID oculto: lo usa el sistema para editar un usuario
existente -->
                <input type="hidden" id="usuarioId" value="">
                <div class="row">
                    <div class="col-md-6 mb-3">
                        <!-- Nombre de usuario: requerido, máximo 16 -->

```

```

        <label for="usuario" class="form-label">Usuario
*=</label>
        <input type="text" class="form-control"
id="usuario" name="usuario"
placeholder="Por favor introduzca un nombre"
required maxlength="16">

        </div>
        <div class="col-md-6 mb-3">
            <!-- Contraseña: validación se explica debajo -->
            <label for="contrasena" class="form-
label">Contraseña *=</label>
            <input type="password" class="form-control"
id="contrasena" name="contrasena" required
maxlength="16">
            <!-- Reglas de contraseña para el usuario -->
            <div class="form-text">
                La contraseña debe tener entre 6 y 16
caracteres, al menos 1 letra mayúscula, 1 número y 1
carácter especial (ej: !, @, #, $).
            </div>
        </div>
        </div>
        <div class="row">
            <div class="col-md-6 mb-3">
                <!-- Rol del usuario: define permisos -->
                <label for="rol" class="form-label">Rol *</label>
                <select class="form-select" id="rol" name="rol"
required>
                    <option value="">Seleccione un rol</option>
                    <option
value="Administrador">ADMINISTRADOR</option>
                    <option value="Salonero">SALONERO</option>
                    <option
value="Propietario">PROPIETARIO</option>
                </select>
            </div>
            <div class="col-md-6 mb-3">
                <!-- Descripción de privilegios: solo lectura, se
llena según el rol -->
                <label for="privilegios" class="form-
label">Privilegios (descripción)</label>
                <textarea class="form-control" id="privilegios"
name="privilegios" rows="3" readonly></textarea>
            </div>
        </div>
        <!-- Botones del formulario: limpiar y guardar -->
        <div class="d-flex justify-content-end">

```

```

                <button type="reset" id="cancelBtn" class="btn btn-secondary me-2">Cancelar</button>
                    <button type="submit" id="submitBtn" class="btn btn-primary">Guardar Usuario</button>
                </div>
            </form>
        </div>

        <!-- Lista de usuarios -->
        <div class="card shadow mb-4">
            <div class="card-header usuarios-header d-flex align-items-center gap-3 flex-wrap">
                <h5 class="mb-0 flex-grow-1 titulo-usuarios">Lista de Usuarios</h5>
                <!-- Buscador en tiempo real por nombre de usuario -->
                <input type="text" id="buscadorUsuarios" class="form-control buscador-usuarios"
                    placeholder="Buscar por nombre de usuario..." autocomplete="off">
                <!-- Recargar la lista manualmente -->
                <button class="btn btn-sm btn-outline-light ms-2" onclick="cargarUsuarios()">
                    Actualizar
                </button>
            </div>
            <div class="card-body p-0">
                <div class="table-responsive">
                    <!-- Aquí JS inyecta la tabla con los usuarios -->
                    <div id="usuarioLista" class="p-4 text-center">
                        <!-- Aquí se inyectará la tabla -->
                    </div>
                </div>
            </div>
        </div>
    </div>

    <!-- Script específico de esta vista: maneja crear/editar/listar usuarios -->
    <script src="/CRM_INT/CRM/public/js/Usuario.js?v=<?= time() ?>"></script>

```

|  
| └── .gitignore

```

# 🗑 Archivos de Sistema que NO deben subirse a Git

# === WHATSAPP SESSION DATA (PERSONAL) ===
bridge-node/session/

```

```
bridge-node/.wwebjs_cache/  
bridge-node/.wwebjs_auth/  
bridge-node/daemon/  
  
# === NODE.JS ===  
bridge-node/node_modules/  
npm-debug.log*  
yarn-debug.log*  
yarn-error.log*  
  
# === LOGS ===  
*.log  
logs/  
  
# === BASES DE DATOS LOCALES ===  
*.db  
*.sqlite  
*.sqlite3  
  
# === ARCHIVOS DE CONFIGURACIÓN LOCAL ===  
bridge-node/.env  
config/local.php  
  
# === ARCHIVOS TEMPORALES ===  
*.tmp  
*.temp  
.DS_Store  
Thumbs.db  
  
# === PM2 LOGS ===  
.pm2/  
  
# === BACKUP ARCHIVOS ===  
*.bak  
*.backup  
  
# === ARCHIVOS DE EDITOR ===  
.vscode/  
.idea/  
*.swp  
*.swo  
  
# === ARCHIVOS XAMPP ESPECÍFICOS ===  
.htaccess.local  
php_errors.log
```

|— .htaccess

```
RewriteEngine On  
RewriteBase /CRM_INT/CRM/
```

```
# Evitar múltiples redirecciones  
RewriteCond %{REQUEST_URI} ^/CRM_INT/CRM/CRM_INT/CRM/(.*)$  
RewriteRule ^ /CRM_INT/CRM/%1 [R=301,L]  
  
# Solo aplicar la reescritura si el archivo no existe y no es una petición a  
index.php  
RewriteCond %{REQUEST_FILENAME} !-f  
RewriteCond %{REQUEST_FILENAME} !-d  
RewriteCond %{REQUEST_URI} !^/CRM_INT/CRM/index\.php  
  
RewriteRule ^(.+)$ index.php?view=$1 [QSA,L]
```

|— README.md

```
# CRM Bastos  
--  
--  
  
## Equipo de Desarrollo  
  
Proyecto desarrollado por estudiantes de la **Universidad Técnica  
Nacional (UTN), sede Guanacaste**, de la carrera de **Tecnologías de  
Información**:  
  
- **Christian Paniagua Castro** – Project Manager, Backend, Database  
Manager, Version Control Manager & Frontend  
- **Steven Baltodano Ugarte** – Backend, UX Designer, Frontend  
- **Reyman Barquero Ramírez** –Backend, UX Designer, Frontend  
- **Brayan Vega Ordóñez** –Backend, UX Designer, Frontend  
- **Felipe Sandoval Chaverri** –Backend, Frontend & QA Tester  
  
### Roles principales  
- **Project Manager:** Christian Paniagua Castro  
- **QA (Aseguramiento de calidad):** Felipe Sandoval Chaverri  
- **UX / Experiencia de usuario:** Brayan Vega Ordóñez  
- **Backend:** Christian Paniagua, Steven Baltodano, Reyman Barquero  
- **Frontend:** Steven Baltodano, Brayan Vega  
- **Database Manager (DBA):** Christian Paniagua Castro  
- **Version Control (Git Kraken & Repositorio):** Christian Paniagua  
Castro  
--  
--
```

## ## Descripción del Proyecto

\*\*CRM Bastos\*\* es un sistema de gestión de clientes VIP para restaurantes. Su propósito es mejorar la fidelización de clientes, optimizar la comunicación y brindar herramientas de análisis que apoyen la toma de decisiones del negocio.

Este sistema permite automatizar recordatorios, gestionar beneficios personalizados, llevar control de compras y códigos de barras, organizar tareas del personal y mantener un historial de acciones dentro del restaurante. Está diseñado para ser utilizado por **administradores, propietarios y personal de salón**, con permisos y vistas diferenciadas para cada rol.

---

## ## Arquitectura y Buenas Prácticas

- **Patrón MVC (Modelo-Vista-Controlador)** para separación de responsabilidades.
  - Uso de **DAO (Data Access Object)**, **DTO (Data Transfer Object)** y **Mapper** para acceso, transporte y mapeo de datos.
  - **Procedimientos almacenados en MySQL** para mayor seguridad y eficiencia.
  - Código modular, limpio y escalable.
- 

## ## Módulos Principales

- **Clients VIP:** gestión, validación y búsqueda de clientes.
  - **Cumpleaños:** listado semanal, envío de correos, historial de felicitaciones y cambio automático de estado.
  - **Compras y Códigos de Barra:** registro de transacciones, emisión y lectura de códigos para promociones.
  - **Tareas (Dashboard):** tarjetas tipo libreta, estados pendientes/completadas, orden cronológico y buscador.
  - **Bitácora:** registro de eventos y operaciones para trazabilidad.
  - **Análisis:** panel con métricas visuales sobre clientes y actividad.
  - **Usuarios y Sesiones:** roles diferenciados (Administrador / Salónero).
- 

## ## Tecnologías Utilizadas

- **Backend:** PHP 8+, MySQL
- **Frontend:** HTML5, CSS3, JavaScript (jQuery), Bootstrap
- **Correo:** PHPMailer

- **IDE recomendado:** Visual Studio Code
  - **Control de versiones:** GitKraken + GitHub
  - **Integración con WhatsApp:** Node.js (bridge-node con `node-windows`)
- 
- 

|— README\_WHATSAPP.md

```
# 📁 CRM con Integración WhatsApp

## 🔍 Setup Inicial
### Rutas para que funcionen los comandos
- **Windows:** `C:\xampp\htdocs\CRM_INT\CRM\bridge-node`

### 1. Instalar Dependencias Node.js
```bash
# Desde la raíz del proyecto CRM
cd bridge-node
npm install
```

# Desde la raíz del proyecto CRM aca es donde les va a mandar el link
para escanear el qr
```

## 🔑 Instalación como Servicio de Windows (Método Definitivo)

### Paso 1: Instalar node-windows
**Requiere PowerShell/CMD como Administrador**
```bash
npm install -g node-windows
```

### Paso 2: Navegar al directorio del bridge
```bash
cd C:\xampp\htdocs\CRM_INT\CRM\bridge-node
```

### Paso 3: Instalar node-windows localmente
```bash
npm install node-windows
```

### Paso 4: Ejecutar el instalador del servicio
```bash
node install-service.cjs
```

```

```

```
** El script automáticamente:** 
- Instala el servicio "WhatsApp Bridge CRM"
- Lo inicia automáticamente
- Lo configura para arrancar con Windows
- ¡Ya no necesitas mantener CMD abierto!

### Paso 5: Verificar que funciona
```bash
curl http://localhost:3001/status-json
```

**Respuesta esperada:** `{"ready":false,"qr":null}` (luego hacer login de WhatsApp)

## 🔍 Ventajas del Servicio de Windows

 **Funciona independientemente de CMD/PowerShell**
 **Inicia automáticamente con Windows**
 **No se cierra al cerrar la consola**
 **Reinicio automático en caso de crash**
 **Gestión desde Services de Windows**
 **Logs automáticos del sistema**

## 📈 Gestión del Servicio

### Verificar estado del servicio
```bash
sc query "WhatsApp Bridge CRM"
```

### Parar el servicio
```bash
sc stop "WhatsApp Bridge CRM"
```

### Iniciar el servicio
```bash
sc start "WhatsApp Bridge CRM"
```

### Desinstalar el servicio (si es necesario)
**Crear archivo uninstall-service.cjs:** 
```javascript
var Service = require('node-windows').Service;

var svc = new Service({
```

```

```

name: 'WhatsApp Bridge CRM',
script: 'C:\\xampp\\htdocs\\CRM_INT\\CRM\\bridge-node\\server.js'
});

svc.on('uninstall', function(){
  console.log('☒ Servicio WhatsApp Bridge desinstalado');
});

svc.uninstall();
```

```

**\*\*Ejecutar:\*\*** `node uninstall-service.cjs`

### ## 📈 Solución de Problemas

```

#### Si el servicio no inicia:
1. **Verificar permisos de administrador**
2. **Revisar logs del servicio en Event Viewer**
3. **Reinstalar el servicio:**
```
bash
node uninstall-service.cjs
node install-service.cjs
```

```

### └─ index.php

```

<?php
session_start();

// Cierre de sesión (logout)
if (isset($_GET['logout'])) {
    session_unset();
    session_destroy();
    header("Location: index.php");
    exit;
}

// Expira por inactividad (30 min)
if (isset($_SESSION['login_time']) && (time() - $_SESSION['login_time']) > 1800) {
    session_unset();
    session_destroy();
    header("Location: index.php");
    exit;
}

```

```
// Verificar autenticación para TODAS las vistas
if (!isset($_SESSION['authenticated']) || $_SESSION['authenticated'] !==
true) {
    include_once 'view/LoginView.php';
    exit; // Detener ejecución si no está autenticado
}

// Refrescar tiempo activo
$_SESSION['login_time'] = time();

// Cargar la vista solicitada (o dashboard por defecto)
$view = $_GET['view'] ?? 'dashboard';
require_once "view/LayoutView.php";
```

## Script de la bd

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Aug 14, 2025 at 05:09 PM
-- Server version: 10.4.32-MariaDB
-- PHP Version: 8.2.12

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

-- Database: `bd_bastos_crm`


DELIMITER $$

-- Procedures
-- 

CREATE DEFINER=`root`@`localhost` PROCEDURE `BitacoraCreate` (IN `p_IdUsuario` INT, IN `p_HoraEntrada` TIME, IN `p_HoraSalida` TIME, IN `p_Fecha` DATE) BEGIN
    INSERT INTO bitacora (IdUsuario, HoraEntrada, HoraSalida, Fecha)
    VALUES (p_IdUsuario, p_HoraEntrada, p_HoraSalida, p_Fecha);
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `BitacoraRead` (IN `p_Id` INT) BEGIN
    SELECT * FROM bitacora WHERE Id = p_Id;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `BitacoraReadAll` () BEGIN
    SELECT * FROM bitacora;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `BitacoraSelectByUsuario` (IN `p_IdUsuario` INT) BEGIN
    SELECT * FROM bitacora WHERE IdUsuario = p_IdUsuario;
```

```

END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `BitacoraUpdate` (IN `p_Id` INT, IN `p_IdUsuario` INT, IN `p_HoraEntrada` TIME, IN `p_HoraSalida` TIME, IN `p_Fecha` DATE) BEGIN
    UPDATE bitacora
    SET IdUsuario = p_IdUsuario,
        HoraEntrada = p_HoraEntrada,
        HoraSalida = p_HoraSalida,
        Fecha = p_Fecha
    WHERE Id = p_Id;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `ClienteCreate` (IN `pCedula` VARCHAR(20), IN `pNombre` VARCHAR(120), IN `pCorreo` VARCHAR(120), IN `pTelefono` VARCHAR(30), IN `pLugarResidencia` VARCHAR(200), IN `pFechaCumpleanos` DATE, IN `pAlergias` TEXT, IN `pGustosEspeciales` TEXT) BEGIN
    INSERT INTO cliente (
        Cedula, Nombre, Correo, Telefono, LugarResidencia,
        FechaCumpleanos,
        Alergias, gustos_especiales
    ) VALUES (
        pCedula, pNombre, pCorreo, pTelefono, pLugarResidencia,
        pFechaCumpleanos,
        pAlergias, pGustosEspeciales
    );
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `ClienteCumpleSemana`()
BEGIN
    DECLARE hoy DATE;
    DECLARE domingo DATE;

    SET hoy = DATE_SUB(CURDATE(), INTERVAL (DAYOFWEEK(CURDATE()) + 5) % 7 DAY);
    SET domingo = DATE_ADD(hoy, INTERVAL 6 DAY);

    SET @hoy := hoy;
    SET @domingo := domingo;

    SELECT
        c.Id,
        c.Cedula,
        c.Nombre,
        c.Correo,
        c.Telefono,
        c.FechaCumpleanos

```

```

    FROM cliente c
    WHERE
        DATE_FORMAT(c.FechaCumpleanos, '%m-%d') BETWEEN DATE_FORMAT(@hoy,
        '%m-%d') AND DATE_FORMAT(@domingo, '%m-%d')
        AND c.Id NOT IN (
            SELECT cu.IdCliente
            FROM cumple cu
            WHERE cu.FechaLlamada BETWEEN @hoy AND @domingo
        );
    END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `ClienteCumpleSemanaOffset`
(IN `p_offset` INT) BEGIN
    DECLARE lunes DATE;
    DECLARE domingo DATE;

    -- Lunes de la semana actual + (7 * offset) días
    SET lunes = DATE_ADD(
        DATE_SUB(CURDATE(), INTERVAL (DAYOFWEEK(CURDATE()) +
5) % 7 DAY),
        INTERVAL 7 * p_offset DAY
    );
    SET domingo = DATE_ADD(lunes, INTERVAL 6 DAY);

    SELECT
        c.Id,
        c.Cedula,
        c.Nombre,
        c.Correo,
        c.Telefono,
        c.FechaCumpleanos
    FROM cliente c
    WHERE DATE_FORMAT(c.FechaCumpleanos, '%m-%d') IN (
        DATE_FORMAT(lunes, '%m-%d'),
        DATE_FORMAT(DATE_ADD(lunes, INTERVAL 1 DAY), '%m-%d'),
        DATE_FORMAT(DATE_ADD(lunes, INTERVAL 2 DAY), '%m-%d'),
        DATE_FORMAT(DATE_ADD(lunes, INTERVAL 3 DAY), '%m-%d'),
        DATE_FORMAT(DATE_ADD(lunes, INTERVAL 4 DAY), '%m-%d'),
        DATE_FORMAT(DATE_ADD(lunes, INTERVAL 5 DAY), '%m-%d'),
        DATE_FORMAT(DATE_ADD(lunes, INTERVAL 6 DAY), '%m-%d')
    )
    AND c.Id NOT IN (
        SELECT cu.IdCliente
        FROM cumple cu
        WHERE cu.FechaLlamada BETWEEN lunes AND domingo
    )
    ORDER BY MONTH(c.FechaCumpleanos), DAY(c.FechaCumpleanos), c.Nombre;
END$$

```

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `ClienteDelete` (IN `pId` INT) BEGIN
    DELETE FROM cliente WHERE Id = pId;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `ClienteExisteCedula` (IN `p_Cedula` VARCHAR(20), IN `p_Id` INT) BEGIN
    SELECT Id
    FROM cliente
    WHERE Cedula = p_Cedula AND (p_Id IS NULL OR Id != p_Id)
    LIMIT 1;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `ClienteRead` (IN `pId` INT) BEGIN
    SELECT
        Id,
        Cedula,
        Nombre,
        Correo,
        Telefono,
        LugarResidencia,
        FechaCumpleanos,
        Acumulado,
        FechaRegistro,
        Alergias AS Alergias,
        gustos_especiales AS GustosEspeciales,
        TotalHistorico -- deja lo que ya tenías
    FROM cliente
    WHERE Id = pId;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `ClienteReadAll` () BEGIN
    SELECT
        Id,
        Cedula,
        Nombre,
        Correo,
        Telefono,
        LugarResidencia,
        FechaCumpleanos,
        Acumulado,
        FechaRegistro,
        Alergias AS Alergias,
        gustos_especiales AS GustosEspeciales,
        TotalHistorico
    FROM cliente

```

```

        ORDER BY Id DESC;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `ClienteUpdate` (IN `pId` INT, IN `pCedula` VARCHAR(20), IN `pNombre` VARCHAR(120), IN `pCorreo` VARCHAR(120), IN `pTelefono` VARCHAR(30), IN `pLugarResidencia` VARCHAR(200), IN `pFechaCumpleanos` DATE, IN `pAcumulado` DECIMAL(10,2), IN `pAlergias` TEXT, IN `pGustosEspeciales` TEXT)    BEGIN
    UPDATE cliente
    SET Cedula = pCedula,
        Nombre = pNombre,
        Correo = pCorreo,
        Telefono = pTelefono,
        LugarResidencia = pLugarResidencia,
        FechaCumpleanos = pFechaCumpleanos,
        Acumulado = pAcumulado,
        Alergias = pAlergias,
        gustos_especiales = pGustosEspeciales
    WHERE Id = pId;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `ClienteVerificarDuplicados` (IN `p_Telefono` VARCHAR(20), IN `p_Correo` VARCHAR(100), IN `p_Id` INT)    BEGIN
    SELECT Id FROM cliente
    WHERE
        (Telefono = p_Telefono OR
        (p_Correo IS NOT NULL AND p_Correo != '' AND Correo = p_Correo))
        AND (Id != p_Id OR p_Id IS NULL);
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `CodigoCreate` (IN `p_IdCliente` INT, IN `p_CodigoBarra` VARCHAR(64), IN `p_CantImpresiones` INT)    BEGIN
    INSERT INTO codigo (IdCliente, CodigoBarra, CantImpresiones)
    VALUES (p_IdCliente, p_CodigoBarra, p_CantImpresiones);
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `CodigoDelete` (IN `p_Id` INT)    BEGIN
    DELETE FROM codigo WHERE Id = p_Id;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `CodigoRead` (IN `p_Id` INT)    BEGIN
    SELECT * FROM codigo WHERE Id = p_Id;
END$$

```

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `CodigoReadAll` ()    BEGIN
    SELECT * FROM codigo;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `CodigoReassign` (IN
`p_IdCliente` INT, IN `p_Motivo` VARCHAR(100))    BEGIN
    DECLARE codActual VARCHAR(64);
    DECLARE codActualExiste INT;
    DECLARE codNuevo VARCHAR(64);
    DECLARE intento INT DEFAULT 0;
    -- Obtener código actual activo del cliente
    SELECT CódigoBarra INTO codActual
    FROM codigo
    WHERE IdCliente = p_IdCliente AND Estado = 'Activo'
    ORDER BY Id DESC
    LIMIT 1;
    -- Verificar que existe
    IF codActual IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'El cliente no tiene una tarjeta activa para
reasignar.';
    END IF;
    -- Generar nuevo código único
    generar_codigo: LOOP
        SET codNuevo = CONCAT('VIP', LPAD(FLOOR(RAND() * 99999), 5,
'0'));
        IF NOT EXISTS (SELECT 1 FROM codigo WHERE CódigoBarra = codNuevo)
THEN
            LEAVE generar_codigo;
        END IF;
        SET intento = intento + 1;
        IF intento > 10 THEN
            SIGNAL SQLSTATE '45001'
            SET MESSAGE_TEXT = 'No se pudo generar un código único.';
        END IF;
    END LOOP;
    -- Desactivar código anterior con motivo y fecha
    UPDATE codigo
    SET Estado = 'Inactivo',
        MotivoCambio = p_Motivo,
        FechaCambio = NOW()
    WHERE CódigoBarra = codActual AND IdCliente = p_IdCliente;
    -- Crear nuevo código usando procedimiento existente
    CALL CódigoCreate(p_IdCliente, codNuevo, 1);
    -- Retornar nuevo código (opcional)
    SELECT codNuevo AS NuevoCodigoGenerado;
END$$

```

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `CodigoSelectByCliente` (IN
`p_IdCliente` INT)    BEGIN
    SELECT * FROM codigo WHERE IdCliente = p_IdCliente;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `CodigoUpdate` (IN `p_Id` INT,
IN `p_CodigoBarra` VARCHAR(64), IN `p_CantImpresiones` INT)    BEGIN
    UPDATE codigo
    SET CodigoBarra = p_CodigoBarra,
        CantImpresiones = p_CantImpresiones
    WHERE Id = p_Id;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `CompraCreate` (IN
`p_FechaCompra` DATE, IN `p_Total` INT, IN `p_IdCliente` INT)    BEGIN
    INSERT INTO compra (FechaCompra, Total, IdCliente)
    VALUES (p_FechaCompra, p_Total, p_IdCliente);
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `CompraDelete` (IN
`p_IdCompra` INT)    BEGIN
    DELETE FROM compra WHERE IdCompra = p_IdCompra;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `CompraRead` (IN `p_IdCompra` INT)    BEGIN
    SELECT * FROM compra WHERE IdCompra = p_IdCompra;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `CompraReadAll` ()    BEGIN
    SELECT * FROM compra;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `CompraReadByCliente` (IN
`p_IdCliente` INT)    BEGIN
    SELECT * FROM compra WHERE IdCliente = p_IdCliente;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `CompraUpdate` (IN
`p_IdCompra` INT, IN `p_FechaCompra` DATE, IN `p_Total` INT, IN
`p_IdCliente` INT)    BEGIN
    UPDATE compra
    SET FechaCompra = p_FechaCompra,
        Total = p_Total,
        IdCliente = p_IdCliente
    WHERE IdCompra = p_IdCompra;
END$$

```

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `PurgarHistorialCumple30Dias`
()    BEGIN
      DELETE FROM cumple
      WHERE FechaLlamada < (NOW() - INTERVAL 30 DAY);
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `SumarCompra` (IN
`p_IdCliente` INT, IN `p_Monto` INT)    BEGIN
    /* Manejador de errores: si algo falla ⇒ ROLLBACK */
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
    END;
    START TRANSACTION;
    /* 1) Registrar la compra */
    INSERT INTO compra (FechaCompra, Total, IdCliente)
    VALUES (NOW(), p_Monto, p_IdCliente);
    /* 2) Sumar al saldo y al histórico */
    UPDATE cliente
    SET Acumulado      = IFNULL(Acumulado, 0)      + p_Monto,
        TotalHistorico = IFNULL(TotalHistorico, 0) + p_Monto
    WHERE Id = p_IdCliente;
    COMMIT;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `TareaCreate` (IN
`p_descripcion` VARCHAR(220), IN `p_estado` 
ENUM('pendiente','completado'))    BEGIN
    INSERT INTO tarea (descripcion, estado)
    VALUES (p_descripcion, p_estado);
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `TareaDelete` (IN `p_Id` 
INT)    BEGIN
    DELETE FROM tarea WHERE id = p_Id;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `TareaRead` (IN `p_Id` 
INT)    BEGIN
    SELECT * FROM tarea WHERE id = p_Id;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `TareaReadAll` ()    BEGIN
    SELECT * FROM tarea ORDER BY fecha_creacion DESC;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `TareaUpdate` (IN `p_Id` INT,
IN `p_Estado` ENUM('pendiente','completada'))    BEGIN

```

```

UPDATE Tarea
SET estado = p_Estado
WHERE id = p_Id;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `UsuarioCreate` (IN
`p_Usuario` VARCHAR(25), IN `p_Contrasena` VARCHAR(255), IN `p_Rol` 
VARCHAR(20)) BEGIN
    INSERT INTO usuario (Usuario, Contrasena, Rol)
    VALUES (p_Usuario, p_Contrasena, p_Rol);
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `UsuarioDelete` (IN `p_Id` 
INT) BEGIN
    DELETE FROM usuario WHERE Id = p_Id;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `UsuarioRead` (IN `p_Id` 
INT) BEGIN
    SELECT * FROM usuario WHERE Id = p_Id;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `UsuarioReadAll` () BEGIN
    SELECT * FROM usuario;
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `UsuarioUpdate` (IN `p_Id` 
INT, IN `p_Usuario` VARCHAR(25), IN `p_Contrasena` VARCHAR(255), IN 
`p_Rol` VARCHAR(20)) BEGIN
    UPDATE usuario
    SET Usuario = p_Usuario,
        Contrasena = p_Contrasena,
        Rol = p_Rol
    WHERE Id = p_Id;
END$$

DELIMITER ;

-----
-- Table structure for table `bitacora`
--

CREATE TABLE `bitacora` (
`IdUsuario` int(11) NOT NULL,
`HoraEntrada` time NOT NULL,
`HoraSalida` time DEFAULT NULL,

```

```

`Fecha` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- 
-- Dumping data for table `bitacora`
-- 

INSERT INTO `bitacora` (`IdUsuario`, `HoraEntrada`, `HoraSalida`,
`Fecha`) VALUES
(1005, '20:04:38', '21:09:03', '2025-07-15'),
(1005, '21:10:32', '20:58:25', '2025-07-15'),
(1005, '21:46:58', '20:46:32', '2025-07-15'),
(1005, '21:48:00', '16:45:58', '2025-07-15'),
(1005, '16:35:20', '16:44:31', '2025-07-16'),
(1005, '21:03:45', '16:41:24', '2025-07-16'),
(1005, '21:51:09', '16:13:33', '2025-07-16'),
(1005, '21:52:55', '15:59:40', '2025-07-16'),
(1015, '15:59:55', '16:00:05', '2025-07-28'),
(1016, '16:00:14', '16:00:45', '2025-07-28'),
(1017, '16:01:00', '16:12:46', '2025-07-28'),
(1017, '16:41:34', '16:41:52', '2025-07-28'),
(1015, '16:42:04', '16:42:28', '2025-07-28'),
(1018, '16:44:43', '16:44:56', '2025-07-28'),
(1018, '16:46:15', '16:47:20', '2025-07-28'),
(1015, '21:50:22', '22:05:54', '2025-07-28'),
(1005, '21:09:08', '21:10:07', '2025-08-06'),
(1005, '21:10:12', '21:13:06', '2025-08-06'),
(1005, '21:13:11', '21:14:51', '2025-08-06'),
(1005, '21:14:56', '21:32:33', '2025-08-06'),
(1005, '21:32:39', '22:08:34', '2025-08-06'),
(1005, '22:08:38', '23:36:05', '2025-08-06'),
(1005, '23:36:16', '21:05:42', '2025-08-06'),
(1005, '21:23:04', '22:50:03', '2025-08-13'),
(1005, '08:39:07', '00:00:00', '2025-08-14');

-----


-- 
-- Table structure for table `cliente`
-- 

CREATE TABLE `cliente` (
  `Id` int(4) NOT NULL,
  `Cedula` varchar(14) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
  `Nombre` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,

```

```

`Correo` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci
DEFAULT NULL,
`Telefono` varchar(8) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci
NOT NULL,
`LugarResidencia` varchar(22) CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci NOT NULL,
`FechaCumpleanos` date NOT NULL,
`Acumulado` int(6) DEFAULT NULL,
`FechaRegistro` date NOT NULL,
`TotalHistorico` int(9) DEFAULT NULL,
`alergias` text DEFAULT NULL,
`gustos_especiales` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

-- 
-- Dumping data for table `cliente`
-- 

INSERT INTO `cliente` (`Id`, `Cedula`, `Nombre`, `Correo`, `Telefono`,
`LugarResidencia`, `FechaCumpleanos`, `Acumulado`, `FechaRegistro`,
`TotalHistorico`, `alergias`, `gustos_especiales`) VALUES
(1, '', 'Vivana Guillen Wong', NULL, '88370518', '', '2024-06-25', NULL,
'2025-08-13', NULL, NULL, NULL),
(2, '502400549', 'Shirley Wong Contreras', NULL, '87671094', '', '1966-
10-04', NULL, '2025-08-13', NULL, NULL, NULL),
(3, '', 'Rody Bogarín Morales', NULL, '83380881', '', '1977-05-17', NULL,
'2025-08-13', NULL, NULL, NULL),
(4, '503930066', 'Diego Wong Salazar', NULL, '85624662', '', '2024-05-
24', NULL, '2025-08-13', NULL, NULL, NULL),
(5, '', 'Guiselle Salazar', NULL, '26690259', '', '2024-05-31', NULL,
'2025-08-13', NULL, NULL, NULL),
(6, '205770879', 'Oscar Vargas Solano', NULL, '88457010', '', '2024-11-
30', NULL, '2025-08-13', NULL, NULL, NULL),
(7, '', 'Ana Mora Gómez', NULL, '83088412', '', '2024-11-28', NULL,
'2025-08-13', NULL, NULL, NULL),
(8, '110260959', 'Evelyn Fallas López', NULL, '88448265', '', '2024-12-
27', NULL, '2025-08-13', NULL, NULL, NULL),
(9, '', 'Ester Solano', NULL, '89237526', '', '2024-07-30', NULL, '2025-
08-13', NULL, NULL, NULL),
(10, '', 'Ana María Solano', NULL, '88191888', '', '0000-00-00', NULL,
'2025-08-13', NULL, NULL, NULL),
(11, '', 'Rebeca Porras', NULL, '84098684', '', '2024-02-01', NULL,
'2025-08-13', NULL, NULL, NULL),
(12, '', 'Marlyn Zeledón', NULL, '', '', '2024-06-27', NULL, '2025-08-
13', NULL, NULL, NULL),
(13, '107560243', 'Carlos Alvarado Ruiz', NULL, '88343096', '', '2024-09-
23', NULL, '2025-08-13', NULL, NULL, NULL),

```

```

(14, '', 'Ander Romero', NULL, '88219158', '', '0000-00-00', NULL, '2025-08-13', NULL, NULL, NULL),
(15, '501640421', 'Ivette Wong Díjeres', NULL, '83832041', '', '2024-10-24', NULL, '2025-08-13', NULL, NULL, NULL),
(16, '503320397', 'Gabriel Alvarez Wong', NULL, '83342718', '', '2024-12-18', NULL, '2025-08-13', NULL, NULL, NULL),
(17, '604210909', 'Jessica Palma Calvo', NULL, '72087806', '', '2024-02-08', NULL, '2025-08-13', NULL, NULL, NULL),
(18, '', 'Carlos Gonzales ', NULL, '25517084', '', '2024-01-13', NULL, '2025-08-13', NULL, NULL, NULL),
(19, '', 'Jose Gonzales', NULL, '11972096', '', '2024-08-26', NULL, '2025-08-13', NULL, NULL, NULL),
(20, '', 'Juan Pablo Mendez Rodríguez', NULL, '87995045', '', '2024-01-10', NULL, '2025-08-13', NULL, NULL, NULL),
(21, '', 'Paola Campos', NULL, '83434686', '', '2024-01-28', NULL, '2025-08-13', NULL, NULL, NULL),
(22, '', 'Celia Apuy', NULL, '83120405', '', '2024-10-05', NULL, '2025-08-13', NULL, NULL, NULL),
(23, '602920117', 'Roque Rojas Pérez', NULL, '89659627', '', '2024-07-07', NULL, '2025-08-13', NULL, NULL, NULL),
(24, '206590043', 'Eduardo Vega Cortes', NULL, '84976306', '', '2024-03-10', NULL, '2025-08-13', NULL, NULL, NULL),
(25, '', 'Lilliana Mora Fanseca', NULL, '83029213', '', '2024-04-17', NULL, '2025-08-13', NULL, NULL, NULL),
(26, '', 'Gabriela Ruiz Salas', NULL, '84422024', '', '2024-08-18', NULL, '2025-08-13', NULL, NULL, NULL),
(27, '', 'Cristhian Gutierrez ', NULL, '88281852', '', '2024-10-19', NULL, '2025-08-13', NULL, NULL, NULL),
(28, '', 'María Fernanda Melendez O.', NULL, '84238294', '', '2024-06-29', NULL, '2025-08-13', NULL, NULL, NULL),
(29, '', 'Luis Andrey Jimenez Alnis', NULL, '89255543', '', '2024-03-02', NULL, '2025-08-13', NULL, NULL, NULL),
(30, '', 'Yecferson Rodríguez Garay', NULL, '88929116', '', '2024-12-22', NULL, '2025-08-13', NULL, NULL, NULL),
(31, '', 'Jessica Vega', NULL, '88458781', '', '2024-05-14', NULL, '2025-08-13', NULL, NULL, NULL),
(32, '', 'Marianela Aguirre', NULL, '89242220', '', '2024-08-09', NULL, '2025-08-13', NULL, NULL, NULL),
(33, '', 'Georgio Jerez Lonza', NULL, '88441313', '', '2024-07-23', NULL, '2025-08-13', NULL, NULL, NULL),
(34, '', 'Darinka ', NULL, '88373584', '', '2024-01-06', NULL, '2025-08-13', NULL, NULL, NULL),
(35, '', 'José Alfredo Tencio', NULL, '83242690', '', '2024-02-15', NULL, '2025-08-13', NULL, NULL, NULL),
(36, '112330284', 'Raquel Murillo Zúñiga', NULL, '87070415', '', '2024-02-02', NULL, '2025-08-13', NULL, NULL, NULL),
(37, '', 'Elizabeth Elizondo Ramírez', NULL, '', '', '2024-02-07', NULL, '2025-08-13', NULL, NULL, NULL),

```

```

(38, '501320086', 'Juan Acón Chen', NULL, '88848607', '', '2024-02-01',
NULL, '2025-08-13', NULL, NULL, NULL),
(39, '', 'José Rojas Alvarez', NULL, '83211089', '', '2024-02-15', NULL,
'2025-08-13', NULL, NULL, NULL),
(40, '', 'KriSttel Fernández', NULL, '83766421', '', '2024-08-03', NULL,
'2025-08-13', NULL, NULL, NULL),
(41, '', 'Odalis Fernández', NULL, '83798400', '', '2024-08-05', NULL,
'2025-08-13', NULL, NULL, NULL),
(42, '', 'Luis Lacayo', NULL, '72065198', '', '2024-07-19', NULL, '2025-
08-13', NULL, NULL, NULL),
(43, '', 'Felipe Avendaño', NULL, '84354618', '', '2024-06-30', NULL,
'2025-08-13', NULL, NULL, NULL),
(44, '', 'Jalil Tabash Fonseca', NULL, '88601109', '', '2024-04-24',
NULL, '2025-08-13', NULL, NULL, NULL),
(45, '', 'José Pablo Arguedas', NULL, '85099065', '', '2024-03-19', NULL,
'2025-08-13', NULL, NULL, NULL),
(46, '110860262', 'Muriel Grijalba Murillo', NULL, '89934520', '',
'2024-11-06', NULL, '2025-08-13', NULL, NULL, NULL),
(47, '', 'Rafa Mejías', NULL, '88154907', '', '2024-11-04', NULL, '2025-
08-13', NULL, NULL, NULL),
(48, '', 'Euyenit Hernandez', NULL, '88210380', '', '2024-09-08', NULL,
'2025-08-13', NULL, NULL, NULL),
(49, '', 'Dayana Rojas', NULL, '88482979', '', '2024-09-17', NULL, '2025-
08-13', NULL, NULL, NULL),
(50, '', 'Wanda Calderon', NULL, '83131495', '', '2024-09-09', NULL,
'2025-08-13', NULL, NULL, NULL),
(51, '503420930', 'RicardoGutierrez Ramirez', NULL, '88352902', '',
'2024-12-14', NULL, '2025-08-13', NULL, NULL, NULL),
(52, '503700040', 'Johana Bermudez Carvajal', NULL, '85037642', '',
'2024-05-08', NULL, '2025-08-13', NULL, NULL, NULL),
(53, '', 'Italia Valverde', NULL, '83956497', '', '2024-04-25', NULL,
'2025-08-13', NULL, NULL, NULL),
(54, '', 'Andrea Martínez', NULL, '89621393', '', '2024-05-24', NULL,
'2025-08-13', NULL, NULL, NULL),
(55, '', 'Gabriel Guzman', NULL, '86045470', '', '2024-03-22', NULL,
'2025-08-13', NULL, NULL, NULL),
(56, '', 'Francisca Linarez', NULL, '85351796', '', '2024-10-01', NULL,
'2025-08-13', NULL, NULL, NULL),
(57, '', 'Mauricio Castro', NULL, '87017214', '', '2024-09-02', NULL,
'2025-08-13', NULL, NULL, NULL),
(58, '', 'Erick Bastos', NULL, '88357772', '', '2024-02-18', NULL, '2025-
08-13', NULL, NULL, NULL),
(59, '502420006', 'Daniel Bastos Luna', NULL, '', '', '2024-06-27', NULL,
'2025-08-13', NULL, NULL, NULL),
(60, '111010528', 'Mariana Rovira Apuy', NULL, '89924694', '', '2024-04-
22', NULL, '2025-08-13', NULL, NULL, NULL),
(61, '401690075', 'Andrey Rojas Méndez', NULL, '88129729', '', '2024-12-
29', NULL, '2025-08-13', NULL, NULL, NULL),

```

```

(62, '', 'Scarlet Vargas Picado', NULL, '64376961', '', '2024-04-01',
NULL, '2025-08-13', NULL, NULL, NULL),
(63, '', 'Raquel Gamboa Garcia', NULL, '88409227', '', '2024-07-05',
NULL, '2025-08-13', NULL, NULL, NULL),
(64, '', 'Ivannia Campos Araya', NULL, '84854528', '', '2024-08-12',
NULL, '2025-08-13', NULL, NULL, NULL),
(65, '', 'Senia Barrios', NULL, '89290404', '', '2024-10-24', NULL,
'2025-08-13', NULL, NULL, NULL),
(66, '124300985', 'Geiner Jose Murillo Castro', NULL, '89787612', '',
'0000-00-00', NULL, '2025-08-13', NULL, NULL, NULL),
(67, '', 'Eduardo Espinal', NULL, '86579608', '', '2024-05-05', NULL,
'2025-08-13', NULL, NULL, NULL);

-- 
-- Triggers `cliente`
-- 

DELIMITER $$

CREATE TRIGGER `InsertCodigoAfterCliente` AFTER INSERT ON `cliente` FOR
EACH ROW BEGIN
    INSERT INTO codigo (IdCliente, CódigoBarra, CantImpresiones)
    VALUES (NEW.Id, CONCAT('CODE-', NEW.Id), 0);
END
$$
DELIMITER ;

-----


-- 
-- Table structure for table `codigo`
-- 

CREATE TABLE `codigo` (
`Id` int(4) NOT NULL,
`IdCliente` int(4) NOT NULL,
`CódigoBarra` varchar(64) DEFAULT NULL,
`CantImpresiones` int(2) DEFAULT 0,
`Estado` enum('Activo','Inactivo') DEFAULT 'Activo',
`MotivoCambio` varchar(50) DEFAULT NULL,
`FechaCambio` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- 
-- Dumping data for table `codigo`
-- 

INSERT INTO `codigo` (`Id`, `IdCliente`, `CódigoBarra`,
`CantImpresiones`, `Estado`, `MotivoCambio`, `FechaCambio`) VALUES
(2499, 1, 'CODE-1', 0, 'Activo', NULL, NULL),

```

```
(2500, 2, 'CODE-2', 0, 'Activo', NULL, NULL),
(2501, 3, 'CODE-3', 0, 'Activo', NULL, NULL),
(2502, 4, 'CODE-4', 0, 'Activo', NULL, NULL),
(2503, 5, 'CODE-5', 0, 'Activo', NULL, NULL),
(2504, 6, 'CODE-6', 0, 'Activo', NULL, NULL),
(2505, 7, 'CODE-7', 0, 'Activo', NULL, NULL),
(2506, 8, 'CODE-8', 0, 'Activo', NULL, NULL),
(2507, 9, 'CODE-9', 0, 'Activo', NULL, NULL),
(2508, 10, 'CODE-10', 0, 'Activo', NULL, NULL),
(2509, 11, 'CODE-11', 0, 'Activo', NULL, NULL),
(2510, 12, 'CODE-12', 0, 'Activo', NULL, NULL),
(2511, 13, 'CODE-13', 0, 'Activo', NULL, NULL),
(2512, 14, 'CODE-14', 0, 'Activo', NULL, NULL),
(2513, 15, 'CODE-15', 0, 'Activo', NULL, NULL),
(2514, 16, 'CODE-16', 0, 'Activo', NULL, NULL),
(2515, 17, 'CODE-17', 0, 'Activo', NULL, NULL),
(2516, 18, 'CODE-18', 0, 'Activo', NULL, NULL),
(2517, 19, 'CODE-19', 0, 'Activo', NULL, NULL),
(2518, 20, 'CODE-20', 0, 'Activo', NULL, NULL),
(2519, 21, 'CODE-21', 0, 'Activo', NULL, NULL),
(2520, 22, 'CODE-22', 0, 'Activo', NULL, NULL),
(2521, 23, 'CODE-23', 0, 'Activo', NULL, NULL),
(2522, 24, 'CODE-24', 0, 'Activo', NULL, NULL),
(2523, 25, 'CODE-25', 0, 'Activo', NULL, NULL),
(2524, 26, 'CODE-26', 0, 'Activo', NULL, NULL),
(2525, 27, 'CODE-27', 0, 'Activo', NULL, NULL),
(2526, 28, 'CODE-28', 0, 'Activo', NULL, NULL),
(2527, 29, 'CODE-29', 0, 'Activo', NULL, NULL),
(2528, 30, 'CODE-30', 0, 'Activo', NULL, NULL),
(2529, 31, 'CODE-31', 0, 'Activo', NULL, NULL),
(2530, 32, 'CODE-32', 0, 'Activo', NULL, NULL),
(2531, 33, 'CODE-33', 0, 'Activo', NULL, NULL),
(2532, 34, 'CODE-34', 0, 'Activo', NULL, NULL),
(2533, 35, 'CODE-35', 0, 'Activo', NULL, NULL),
(2534, 36, 'CODE-36', 0, 'Activo', NULL, NULL),
(2535, 37, 'CODE-37', 0, 'Activo', NULL, NULL),
(2536, 38, 'CODE-38', 0, 'Activo', NULL, NULL),
(2537, 39, 'CODE-39', 0, 'Activo', NULL, NULL),
(2538, 40, 'CODE-40', 0, 'Activo', NULL, NULL),
(2539, 41, 'CODE-41', 0, 'Activo', NULL, NULL),
(2540, 42, 'CODE-42', 0, 'Activo', NULL, NULL),
(2541, 43, 'CODE-43', 0, 'Activo', NULL, NULL),
(2542, 44, 'CODE-44', 0, 'Activo', NULL, NULL),
(2543, 45, 'CODE-45', 0, 'Activo', NULL, NULL),
(2544, 46, 'CODE-46', 0, 'Activo', NULL, NULL),
(2545, 47, 'CODE-47', 0, 'Activo', NULL, NULL),
(2546, 48, 'CODE-48', 0, 'Activo', NULL, NULL),
(2547, 49, 'CODE-49', 0, 'Activo', NULL, NULL),
```

```

(2548, 50, 'CODE-50', 0, 'Activo', NULL, NULL),
(2549, 51, 'CODE-51', 0, 'Activo', NULL, NULL),
(2550, 52, 'CODE-52', 0, 'Activo', NULL, NULL),
(2551, 53, 'CODE-53', 0, 'Activo', NULL, NULL),
(2552, 54, 'CODE-54', 0, 'Activo', NULL, NULL),
(2553, 55, 'CODE-55', 0, 'Activo', NULL, NULL),
(2554, 56, 'CODE-56', 0, 'Activo', NULL, NULL),
(2555, 57, 'CODE-57', 0, 'Activo', NULL, NULL),
(2556, 58, 'CODE-58', 0, 'Activo', NULL, NULL),
(2557, 59, 'CODE-59', 0, 'Activo', NULL, NULL),
(2558, 60, 'CODE-60', 0, 'Activo', NULL, NULL),
(2559, 61, 'CODE-61', 0, 'Activo', NULL, NULL),
(2560, 62, 'CODE-62', 0, 'Activo', NULL, NULL),
(2561, 63, 'CODE-63', 0, 'Activo', NULL, NULL),
(2562, 64, 'CODE-64', 0, 'Activo', NULL, NULL),
(2563, 65, 'CODE-65', 0, 'Activo', NULL, NULL),
(2564, 66, 'CODE-66', 0, 'Activo', NULL, NULL),
(2565, 67, 'CODE-67', 0, 'Activo', NULL, NULL);

-----
-- Table structure for table `compra`


CREATE TABLE `compra` (
  `IdCompra` int(6) NOT NULL,
  `FechaCompra` date NOT NULL,
  `Total` int(6) NOT NULL,
  `IdCliente` int(3) NOT NULL DEFAULT 0
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

-----
-- Table structure for table `cumple`


CREATE TABLE `cumple` (
  `Id` int(11) NOT NULL,
  `IdCliente` int(11) DEFAULT NULL,
  `FechaLlamada` date DEFAULT NULL,
  `Vence` date DEFAULT NULL,
  `Vencido` varchar(5) DEFAULT 'NO'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```



```

`Rol` enum('Administrador','Salonero','Propietario') CHARACTER SET
utf8mb4 COLLATE utf8mb4_general_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

-- 
-- Dumping data for table `usuario`
-- 

INSERT INTO `usuario` (`Id`, `Usuario`, `Contrasena`, `Rol`) VALUES
(1005, 'CHRISTIANGALLO',
'$2y$10$8y6NGmtK8dt.UeYy1kGhNOE4FE4Sh7VkJT637yoUpaOaVAWnrZ/Na',
'Propietario'),
(1006, 'ADMIN',
'$2y$10$1R4g89Q1q/1373ifxsCjPuAzcRGMAkVScmPxatN4AQJCP0KG1GxD6',
'Administrador'),
(1007, 'SALONERO',
'$2y$10$FvFD6TpDE0/cWYeP2jntehlEr8fNzuaa3.yRnJoVgJbvN97YmAbm',
'Salonero'),
(1008, 'PEPEGRILLO',
'$2y$10$b1SqIZ/dRg0/pazFsDKuru1HHKWnnU09H/WOr2stTuBuGPrzuYITq',
'Propietario'),
(1010, '1223',
'$2y$10$73xQfNHQrbOHmNVXayKS4eXYAoJFemXR09uvezF3w0/L5Mc3ppoWe',
'Administrador'),
(1011, 'CHRISTIAN EDUARDO PANIAGU',
'$2y$10$xFhNezxhZ5qfvzYFcLNcBOJvn8M0EuCy50VQP4k1snPVty8UrJrIq',
'Propietario'),
(1012, 'PRUEBA',
'$2y$10$pv5Vc/qmvIJ55cRcXETOieJBcZEp5aoZsV7jxY8Zvz0io8D3dEgMC',
'Salonero'),
(1013, 'PRUEBAA',
'$2y$10$/nBpj22Vrqp2nvqfT.64muW.sVLSiUoVhAnk6TI3Je3k.7knYNdbC',
'Salonero'),
(1014, 'PRUEBAAA',
'$2y$10$sRyChUQVj8Lpx2iPkt20i.6mF0eyLFj/cvXmarOnT670zCxpaNm8G',
'Administrador'),
(1015, 'REYMAN1',
'$2y$10$F2VRB8/qaVjn2e5UfeGc70pnkYMmxjKxikiL8RUdjto4/xudOs1jW',
'Administrador'),
(1016, 'REYMAN2',
'$2y$10$1AcHfWiLBnwbyObM9rQ0fu5iEfMhDM0x1KoCACk/aXfcHEzc7e5S2',
'Salonero'),
(1017, 'REYMAN3',
'$2y$10$K/y5w6RsKRvUJkxay/ZQr.x3xPr/p258QgCgwMEL9rA7UXiJkEE7e',
'Propietario'),
(1018, 'PRUEBADEFINITIVA',
'$2y$10$q5itKcf/xV3BnSHWmaqcdepoai8jc6vcXEtrDtXu1qQMv1eZzJxHO',
'Administrador');

```

```
--  
-- Indexes for dumped tables  
--  
  
--  
-- Indexes for table `bitacora`  
--  
ALTER TABLE `bitacora`  
    ADD KEY `IdUsuario` (`IdUsuario`);  
  
--  
-- Indexes for table `cliente`  
--  
ALTER TABLE `cliente`  
    ADD PRIMARY KEY (`Id`) USING BTREE;  
  
--  
-- Indexes for table `codigo`  
--  
ALTER TABLE `codigo`  
    ADD PRIMARY KEY (`Id`),  
    ADD KEY `IdCliente` (`IdCliente`);  
  
--  
-- Indexes for table `compra`  
--  
ALTER TABLE `compra`  
    ADD PRIMARY KEY (`IdCompra`) USING BTREE,  
    ADD KEY `idCliente` (`IdCliente`) USING BTREE;  
  
--  
-- Indexes for table `cumple`  
--  
ALTER TABLE `cumple`  
    ADD PRIMARY KEY (`Id`),  
    ADD KEY `IdCliente` (`IdCliente`);  
  
--  
-- Indexes for table `tarea`  
--  
ALTER TABLE `tarea`  
    ADD PRIMARY KEY (`id`);  
  
--  
-- Indexes for table `usuario`  
--  
ALTER TABLE `usuario`
```

```

ADD PRIMARY KEY (`Id`);

-- 
-- AUTO_INCREMENT for dumped tables
-- 

-- 
-- AUTO_INCREMENT for table `bitacora`
-- 

ALTER TABLE `bitacora`
  MODIFY `IdUsuario` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=1019;

-- 
-- AUTO_INCREMENT for table `cliente`
-- 

ALTER TABLE `cliente`
  MODIFY `Id` int(4) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=68;

-- 
-- AUTO_INCREMENT for table `codigo`
-- 

ALTER TABLE `codigo`
  MODIFY `Id` int(4) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2566;

-- 
-- AUTO_INCREMENT for table `compra`
-- 

ALTER TABLE `compra`
  MODIFY `IdCompra` int(6) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=28540;

-- 
-- AUTO_INCREMENT for table `cumple`
-- 

ALTER TABLE `cumple`
  MODIFY `Id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=19;

-- 
-- AUTO_INCREMENT for table `tarea`
-- 

ALTER TABLE `tarea`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=78;

-- 
-- AUTO_INCREMENT for table `usuario`
-- 

ALTER TABLE `usuario`
  MODIFY `Id` int(4) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=1019;

```

```

-- 
-- Constraints for dumped tables
-- 

-- 
-- Constraints for table `bitacora`
-- 

ALTER TABLE `bitacora`
  ADD CONSTRAINT `bitacora_ibfk_1` FOREIGN KEY (`IdUsuario`) REFERENCES
`usuario` (`Id`) ON DELETE CASCADE ON UPDATE CASCADE;

-- 
-- Constraints for table `codigo`
-- 

ALTER TABLE `codigo`
  ADD CONSTRAINT `codigo_ibfk_1` FOREIGN KEY (`IdCliente`) REFERENCES
`cliente` (`Id`) ON DELETE CASCADE ON UPDATE CASCADE;

-- 
-- Constraints for table `compra`
-- 

ALTER TABLE `compra`
  ADD CONSTRAINT `compra_ibfk_1` FOREIGN KEY (`IdCliente`) REFERENCES
`cliente` (`Id`) ON DELETE CASCADE ON UPDATE CASCADE;

-- 
-- Constraints for table `cumple`
-- 

ALTER TABLE `cumple`
  ADD CONSTRAINT `cumple_ibfk_1` FOREIGN KEY (`IdCliente`) REFERENCES
`cliente` (`Id`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

## Estructura del Proyecto formato png

```
CRM_INT/
  CRM/
    bridge-node/
      .env.example
      install-service.cjs
      package-lock.json
      package.json
      server.js

    config/
      CorreoExpress.php
      CorreoHelper.php
      Database.php

    controller/
      AnalisisController.php
      BitacoraController.php
      ClienteController.php
      CodigoController.php
      CompraController.php
      CumpleController.php
      ExpressController.php
      MessageController.php
      SessionController.php
      TareaController.php
      UsuarioController.php

    LIB/
      phpmailer/
        Exception.php
        PHPMailer.php
        SMTP.php
        WhatsAppService.php

    model/
      Bitacora/
        BitacoraDAO.php
        BitacoraDTO.php
        BitacoraMapper.php
      cliente/
        ClienteDAO.php
        ClienteDTO.php
        ClienteMapper.php
      codigo/
        CodigoDAO.php
        CodigoDTO.php
        CodigoMapper.php
      compra/
        CompraDAO.php
        CompraDTO.php
        CompraMapper.php
      cumple/
        CumpleDAO.php
        CumpleDTO.php
        CumpleMapper.php
      tarea/
        TareaDAO.php
        TareaDTO.php
        TareaMapper.php
      usuario/
        UsuarioDAO.php
        UsuarioDTO.php
        UsuarioMapper.php

    public/
      css/
        Analisis.css
        Bitacora.css
        Cliente.css
        Codigo.css
        Compra.css
        Cumple.css
        Dashboard.css
        Layout.css
        Login.css
        Manual.css
        Manual2.css
        Usuario.css
      img/
        Amarillo_LBlancas.png
        Amarillo_LNegras.png
        Negro_LNegras.png
        Principal_Amarillo.png
        Principal_Negro.png
      js/
        Analisis.js
        Bitacora.js
        Cliente.js
        Codigo.js
        Compra.js
        Cumple.js
        Dashboard.js
        jquery-3.6.3.min.js
        Layout.js
        Login.js
        Manual.js
        ScannerGlobal.js
        Tarea.js
        Usuario.js

      view/
        AnalisisView.php
        BitacoraView.php
        ClienteView.php
        CodigoView.php
        CompraView.php
        CumpleView.php
        DashboardView.php
        LayoutView.php
        LoginView.php
        ManualView.php
        Manual2View.php
        UsuarioView.php

    .gitignore
    .htaccess
    README.md
    README_WHATSAPP.md
    index.php
```

## Descripción del proyecto

**CRM Bastos** es un sistema de gestión integral diseñado específicamente para restaurantes que buscan fortalecer la relación con sus clientes VIP y optimizar sus procesos internos de atención. El proyecto nace con el propósito de ofrecer una herramienta tecnológica que permita a propietarios, administradores y personal de salón gestionar de manera eficiente la información de los clientes, automatizar procesos de fidelización y disponer de datos estratégicos para la toma de decisiones. El sistema está orientado a crear experiencias personalizadas y memorables para los comensales, incrementando la lealtad hacia el negocio y, en consecuencia, los ingresos recurrentes del restaurante.

La plataforma ha sido desarrollada bajo la metodología **MVC (Modelo-Vista-Controlador)**, garantizando una clara separación de responsabilidades y facilitando el mantenimiento y escalabilidad del software. Cada módulo del sistema sigue las mejores prácticas de programación, integrando patrones de diseño como **DAO (Data Access Object)**, **DTO (Data Transfer Object)** y **Mapper**, los cuales permiten un acceso estructurado a los datos, una transferencia segura de la información y un mapeo coherente entre entidades y bases de datos. Gracias a esta arquitectura, el código mantiene un alto nivel de orden, legibilidad y robustez, asegurando que las futuras ampliaciones o modificaciones se realicen de manera ágil y confiable.

Desde el punto de vista funcional, CRM Bastos integra módulos clave que cubren las principales necesidades de un restaurante en cuanto a gestión de clientes y fidelización. El módulo de **Clientes VIP** permite registrar, actualizar y consultar información validada de cada cliente, evitando duplicidades y centralizando todos los datos relevantes. El módulo de **Cumpleaños** ofrece una gestión semanal de clientes que cumplen años, habilitando el envío automático de correos personalizados y el cambio de estado de cada felicitación, con trazabilidad en un historial visual para el administrador. Por otra parte, el módulo de **Compras y Códigos de Barra** brinda control sobre las transacciones realizadas, la emisión de códigos para promociones y el seguimiento de beneficios aplicados, mientras que el módulo de **Tareas** se presenta en el dashboard como una libreta digital que permite organizar pendientes, marcar avances y dar seguimiento al trabajo operativo de manera sencilla y visual.

El sistema también incluye un módulo de **Bitácora**, en el que se registran las acciones realizadas dentro de la plataforma, garantizando trazabilidad, transparencia y control sobre los procesos internos. Asimismo, dispone de un módulo de **Análisis**, que consolida métricas estratégicas y presenta indicadores de desempeño de los clientes, compras y beneficios, todo ello en un entorno visual e intuitivo que facilita la interpretación de los datos. Los mecanismos de autenticación y control de sesiones están cuidadosamente diseñados para ofrecer seguridad, definiendo roles y permisos según el tipo de usuario: mientras los administradores pueden acceder a todo el ecosistema de funcionalidades, los salones cuentan con vistas simplificadas y adaptadas a su trabajo diario.

En cuanto a su infraestructura tecnológica, el sistema combina la potencia de **PHP 8+**, **MySQL con procedimientos almacenados** y **JavaScript con jQuery**, apoyados en **Bootstrap** para una experiencia responsive y amigable en distintos dispositivos. La arquitectura modular facilita la independencia de cada componente, y la carpeta pública organiza los recursos estáticos de forma clara (CSS, JS, imágenes), asegurando un mantenimiento ágil. Para la mensajería, CRM Bastos incorpora **PHPMailer** para la gestión de correos electrónicos y un servicio de integración con **WhatsApp** a través de un bridge basado en **Node.js**, lo cual amplía las posibilidades de comunicación directa con los clientes.

En síntesis, CRM Bastos no es únicamente un software de gestión, sino una solución estratégica que convierte la información de los clientes en acciones concretas y automatizadas: recordatorios oportunos, beneficios personalizados, comunicación directa y reportes ejecutivos que empoderan al negocio. Está pensado para que el **administrador tenga control total y visión global**, el **propietario cuente con indicadores confiables para la toma de decisiones**, y el **salonero disponga de herramientas simples para identificar a los clientes VIP y brindarles un trato diferenciado**. Todo ello se traduce en un ecosistema organizado, moderno y adaptable, que coloca al cliente en el centro de la operación del restaurante y fortalece la relación entre servicio, tecnología y experiencia gastronómica.

## **Extensiones utilizadas en el desarrollo del sistema**

Language pack extension for Spanish  
PHP Debug  
PHP IntelliSense  
Path Autocomplete  
IntelliSense for CSS  
Prettier  
Bracket Pair Colorizer  
HTML Play  
Material Icon Theme

## **Acceso y Reglas**

Acceso al sistema mediante la ip 10.50.80.218 de la VM del servidor

Usuario administrator

Contraseña de la máquina virtual UTN2025\*\*

Firewall

Reglas en el firewall

puerto	Servicio	Descripción
80	HTTP	Acceso a sistema web desde la red local
3389	RDP	Acceso remoto a la vm

Herramienta de gestión BD: heidisql conectada a localhost

Credenciales para entrar a la base de datos:

Dirección ip 10.50.80.239

Usuario root

Contraseña UTN2025\*\*G2

## API WhatsApp -> Pasos

Rutas de trabajo

**Windows:** C:\xampp\htdocs\CRM\_INT\CRM\bridge-node

1. Instalación de dependencias Node.js

# Desde la raíz del proyecto CRM

cd bridge-node

npm install

# Durante esta instalación, el sistema generará el QR para vincular la cuenta de WhatsApp.

---

Instalación como Servicio de Windows (Método Recomendado)

De esta forma el servicio quedará siempre activo, iniciará con Windows y se reiniciará en caso de fallo.

Paso 1: Instalar node-windows (global)

Ejecutar PowerShell o CMD como **Administrador**

npm install -g node-windows

Paso 2: Navegar al directorio del bridge

cd C:\xampp\htdocs\CRM\_INT\CRM\bridge-node

Paso 3: Instalar node-windows localmente

npm install node-windows

Paso 4: Ejecutar el instalador del servicio

node install-service.cjs

**Este script automáticamente:**

Crea el servicio “**WhatsApp Bridge CRM**”

Lo inicia en segundo plano

Lo configura para arrancar con Windows

Elimina la necesidad de mantener abierta una consola CMD

Paso 5: Verificar estado inicial

curl <http://localhost:3001/status-json>

**Respuesta esperada:**

{"ready":false,"qr":null}

(El estado cambiará tras vincular el WhatsApp con el QR generado)

---

Ventajas de instalarlo como Servicio

Funciona independiente de CMD/PowerShell

Arranca automáticamente junto con Windows

No se detiene al cerrar sesiones de consola  
Reinicio automático en caso de error  
**Administrable desde Servicios de Windows**  
Generación automática de logs del sistema

---

Gestión del Servicio

Verificar estado

sc query "WhatsApp Bridge CRM"

Iniciar servicio

sc start "WhatsApp Bridge CRM"

Detener servicio

sc stop "WhatsApp Bridge CRM"

Desinstalar servicio

Crear archivo uninstall-service.cjs:

```
var Service = require('node-windows').Service;
```

```
var svc = new Service({
```

```
  name: 'WhatsApp Bridge CRM',
```

```
  script: 'C:\\xampp\\htdocs\\CRM_INT\\CRM\\bridge-node\\server.js'
```

```
});
```

```
svc.on('uninstall', function(){
```

```
  console.log('Servicio WhatsApp Bridge desinstalado');
```

```
});
```

```
svc.uninstall();
```

Ejecutar:

```
node uninstall-service.cjs
```

---

Solución de Problemas

Si el servicio no inicia correctamente:

Verificar que se esté ejecutando con **permisos de administrador**.

Revisar los **logs del servicio** en *Event Viewer* de Windows.

Reinstalar el servicio:

```
node uninstall-service.cjs
```

```
node install-service.cjs
```

## Requerimientos de hardware

Para poder utilizar el sistema de forma estable y sin problemas, se recomienda contar con los siguientes equipos:

### 1. Lector de Códigos de Barra

Este dispositivo se utiliza para leer los códigos de las tarjetas o etiquetas de los clientes.

Modelos recomendados:

**Eyoyo EY-007** (aprox. ₡25,000)

**Inateck BCST-70** (aprox. ₡35,000)

### 2. Impresora de Códigos de Barra

Sirve para imprimir las etiquetas que luego se colocan en productos o tarjetas.

Modelo recomendado: **Xprinter XP-350B**, ya que ofrece la mejor relación entre precio y calidad.

Precio aproximado: entre **₡65,000 – ₡80,000**.

Insumos necesarios: **Rollo de etiquetas 40 x 30 mm**, con un costo aproximado de **₡6,000 – ₡8,000** por rollo.

### 3. Computadora Portátil (Laptop)

Es el equipo principal donde funcionará el sistema. Debe cumplir con los siguientes requisitos mínimos:

**Procesador:** Intel® Core™ i5-1145G7

**Velocidad base:** 2.60 GHz (con capacidad de llegar hasta 4.40 GHz en modo turbo)

**Núcleos/Hilos:** 4 núcleos y 8 hilos (permite trabajar en varias tareas a la vez sin lentitud)

**Gráficos integrados:** Intel Iris Xe Graphics (suficiente para manejar el sistema y tareas gráficas básicas)

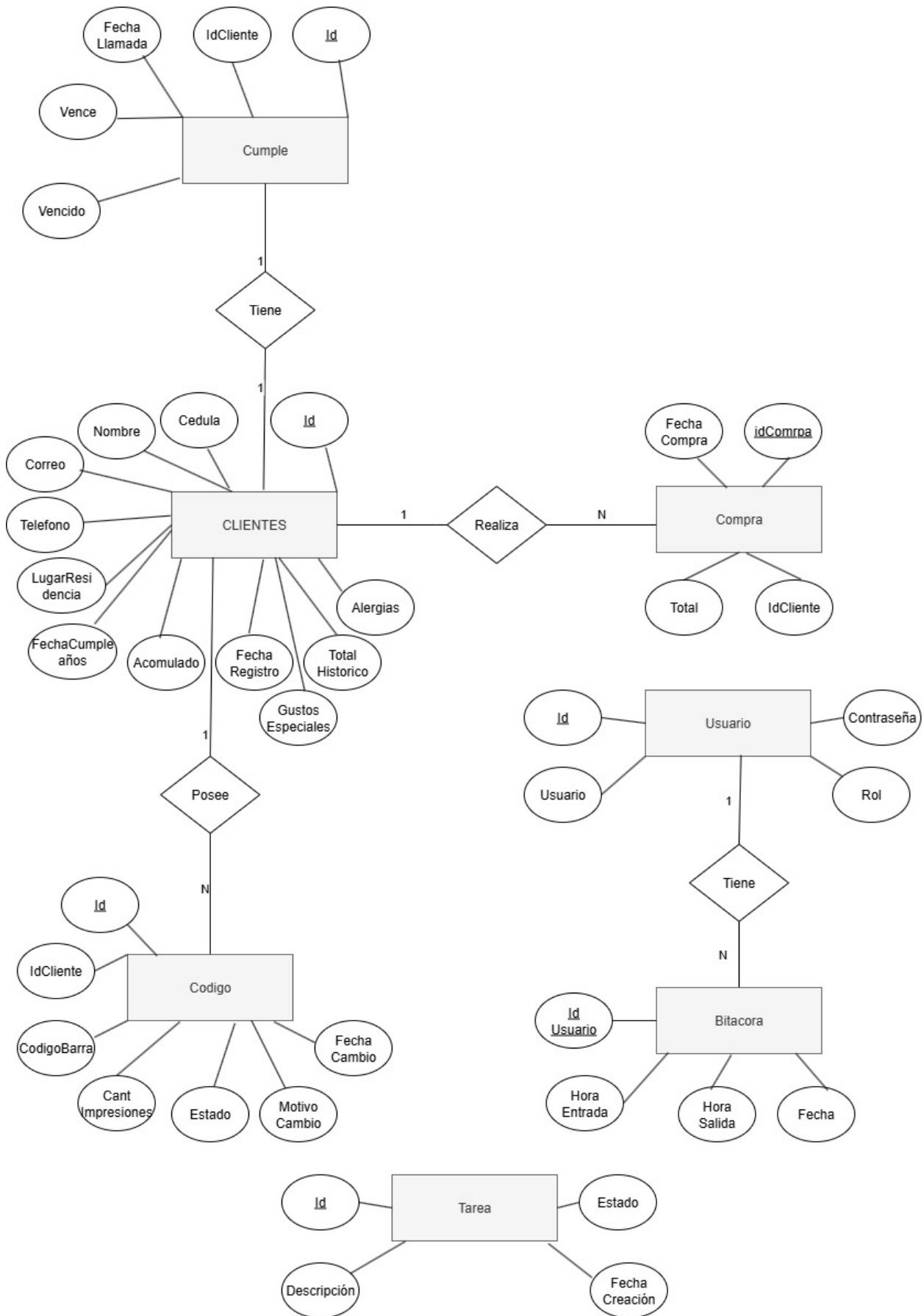
**Memoria RAM:** 16 GB (de los cuales se aprovechan 15.4 GB). Se recomienda **24 GB** para un mejor rendimiento.

**Almacenamiento:** 256 GB SSD (permite rapidez al abrir el sistema y guardar la información).

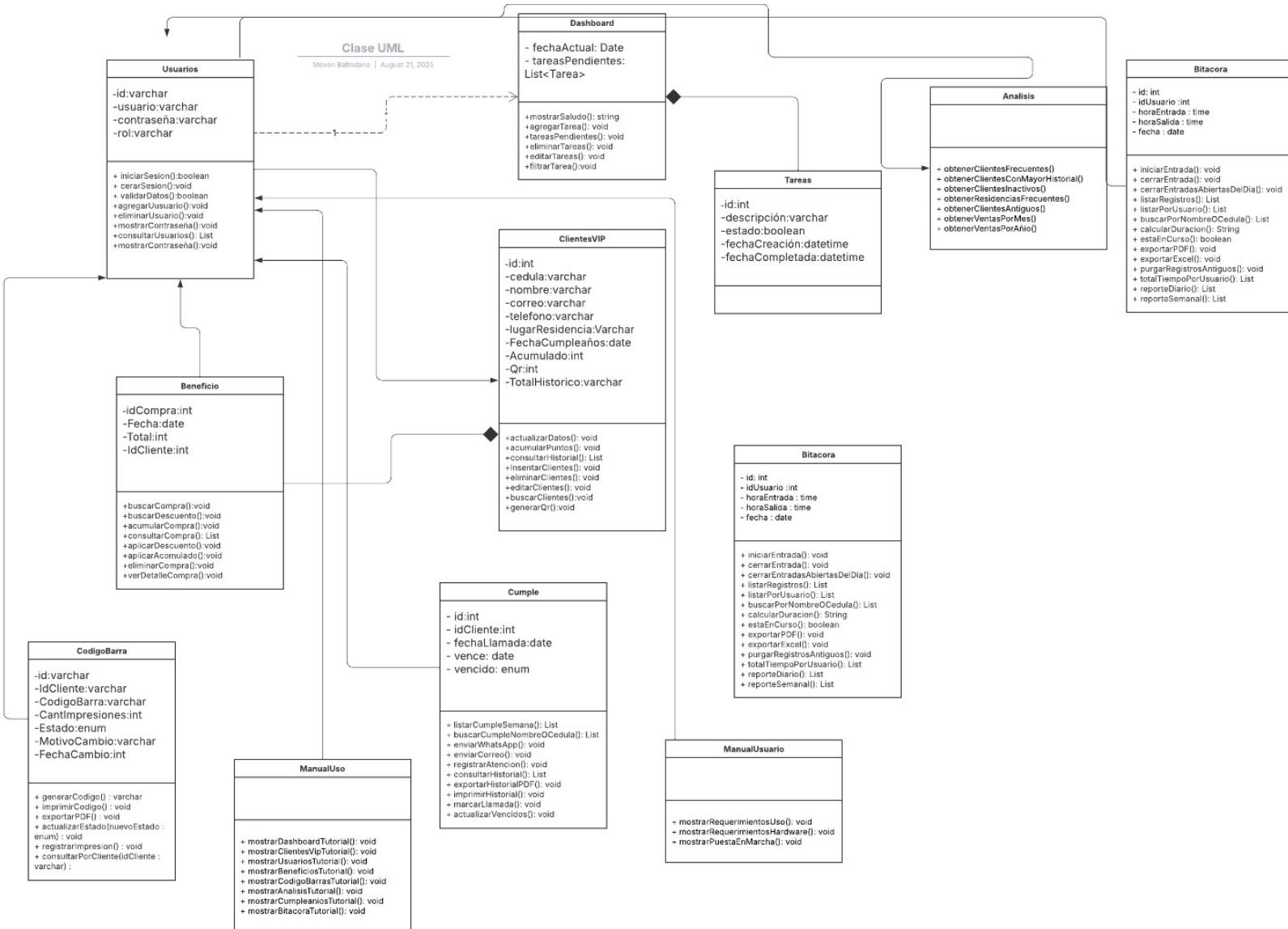
---

Con este hardware, el restaurante podrá trabajar con el CRM de forma fluida, imprimir etiquetas, leer códigos sin errores y garantizar que el sistema funcione de manera ágil y segura.

## Diagrama de entidad-relación



# Diagrama de clases UML



## Diagrama de secuencia Cliente externo

