

Directory Structure:

```
README.md
cadastros.html
consultar-frequencias-turma.html
css/
  style.css
data/
  alunos.json
  presencas.json
  turmas.json
index.html
js/
  controller.js
  model.js
  view.js
marcar-presenca.html
package-lock.json
package.json
relatorio-frequencias-aluno.html
server.js
```

File: consultar-frequencias-turma.html

```
=====

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Consultar Frequências por Turma - Controle de Frequência</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
  <link rel="stylesheet" href="/css/style.css">
</head>
<body>
  <div class="container py-4">
    <h1 class="mb-5 text-center">Consultar Frequências por Turma</h1>
    <div class="card">
      <div class="card-header">Consultar Frequências por Turma</div>
      <div class="card-body">
        <form id="formConsultaFrequencia">
          <div class="row">
            <div class="col-md-6 mb-3">
              <label for="turmaFrequencia" class="form-label">Turma</label>
              <select class="form-select" id="turmaFrequencia" required>
                <option value="" disabled selected>Selecione uma turma</option>
              </select>
            </div>
            <div class="col-md-6 mb-3">
              <label for="dataFrequencia" class="form-label">Data</label>
              <input type="date" class="form-control" id="dataFrequencia" required>
            </div>
          </div>
          <button type="submit" class="btn btn-primary w-100" id="btnConsultarFrequencia">Consultar</button>
        </form>
        <div id="relatorioFrequencia" class="mt-3"></div>
      </div>
    </div>
    <div class="mt-4 text-center">
      <a href="/index.html" class="btn btn-secondary">Voltar ao Menu</a>
    </div>
  </div>
  <!-- Scripts -->
  <script src="/js/model.js"></script>
  <script src="/js/controller.js"></script>
  <script src="/js/view.js"></script>
</body>
</html>
```

File: package-lock.json

```
=====

{
  "name": "controle-frequencia",
  "version": "1.0.0",
  "lockfileVersion": 3,
  "requires": true,
  "packages": {
    "": {
      "name": "controle-frequencia",
      "version": "1.0.0",
      "license": "ISC",
      "dependencies": {
        "cors": "^2.8.5",
        "express": "^5.1.0",
        "uuid": "^11.1.0"
      }
    },
    "node_modules/accepts": {
      "version": "2.0.0",
      "resolved": "https://registry.npmjs.org/accepts/-/accepts-2.0.0.tgz",
      "integrity": "sha512-5e7+eAAB6YwY6vvDMdIvbnGgO9o9Gp+gKqPQWVn34QW1u+qkZjK1Y/RI3Hx4X3Ug183Zn1h1L7620Yg==",
      "dependencies": {
        "mime-types": "^3.0.0",
        "negotiator": "^1.0.0"
      },
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/body-parser": {
      "version": "2.2.0",
      "resolved": "https://registry.npmjs.org/body-parser/-/body-parser-2.2.0.tgz",
      "integrity": "sha512-221I2b2310a1b3+uLz8Jbe09CkOwHKBWVbS7aaJtjS4B8Yz7P6DFBzt13z0w/539X775S4uTtQ7+Nq==",
      "dependencies": {
        "bytes": "^3.1.2",
        "content-type": "^1.0.5",
        "debug": "^4.4.0",
        "http-errors": "^2.0.0",
        "iconv-lite": "^0.6.3",
        "on-finished": "^2.4.1",
        "qs": "^6.14.0",
        "raw-body": "^3.0.0",
        "type-is": "^2.0.0"
      },
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/bytes": {
      "version": "3.1.2",

```

```
"resolved": "https://registry.npmjs.org/bytes/-/bytes-3.1.2.tgz",
"integrity": "sha512-Nf7TyzTx6S3yRJObOAV7956r8cr2+Oj8AC5dt8wSP3BQAoeX58NoHyCU8P8zGkNXStjTSi6fzO6F0pBd
"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/KwFe21cMpmHtPOSij8K99/wSfoEuTObmU
"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/ABpBCqhA9EuxJaQWdQg7295H4hBphv3lZg0boBKuwYpt4YXp6MZ5AmZQnU/tyMT
"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/content-disposition": {
  "version": "1.0.0",
  "resolved": "https://registry.npmjs.org/content-disposition/-/content-disposition-1.0.0.tgz",
  "integrity": "sha512-+ys997U96po4Kx/ABpBCqhA9EuxJaQWdQg7295H4hBphv3lZg0boBKuwYpt4YXp6MZ5AmZQnU/tyMT
"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-+ys997U96po4Kx/ABpBCqhA9EuxJaQWdQg7295H4hBphv3lZg0boBKuwYpt4YXp6MZ5AmZQnU/tyMT
"engines": {
  "node": ">= 0.6"
}
},
"node_modules/cookie": {
```

```
"version": "0.7.2",
"resolved": "https://registry.npmjs.org/cookie/-/cookie-0.7.2.tgz",
"integrity": "sha512-yki5XnKuf750l50uGTlt6kKILY4nQ1eNIQatoXEBYz5dWgnKqbnqmTrBE5B4N7lrMJKQ2ytWMiTO2o0v6E",
"engines": {
  "node": ">= 0.6"
}
},
"node_modules/cookie-signature": {
  "version": "1.2.2",
  "resolved": "https://registry.npmjs.org/cookie-signature/-/cookie-signature-1.2.2.tgz",
  "integrity": "sha512-D76uU73uISXrD1UXF4KE2TMxVVwhsnCgfAyTg9k8P6KGZjlXKrOLe4dJQKI3Bxi5wjEsZoFXJWEINWBJF",
  "engines": {
    "node": ">=6.6.0"
  }
},
"node_modules/cors": {
  "version": "2.8.5",
  "resolved": "https://registry.npmjs.org/cors/-/cors-2.8.5.tgz",
  "integrity": "sha512-KIHbLJqu73RGr/hnbrO9uBeixNGuvSQjul/jdFvS/KFSIH1hWVd1ng7zOHx+YrEflnLG7q4n6GHQ9cDtxv/P",
  "dependencies": {
    "object-assign": "^4",
    "vary": "^1"
  },
  "engines": {
    "node": ">= 0.10"
  }
},
"node_modules/debug": {
  "version": "4.4.0",
  "resolved": "https://registry.npmjs.org/debug/-/debug-4.4.0.tgz",
  "integrity": "sha512-6WTZ/lxCY/T6BALoZHaE4ctp9xm+Z5kY/pzYaCHRFeyVhojxlrM+46y68HA6hr0TcwEssoxNiDEUJQjfPZ/",
  "dependencies": {
    "ms": "^2.1.3"
  },
  "engines": {
    "node": ">=6.0"
  }
},
"peerDependenciesMeta": {
  "supports-color": {
    "optional": true
  }
}
},
"node_modules/depd": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/depd/-/depd-2.0.0.tgz",
  "integrity": "sha512-g7NShpUXUEGr1sZuLcXwvU5DmiWJl66FtnK3wFozgXjI8tZCiw59oPNL1oQ2k0vR7bQI88/5Z3XN5320g=",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/dunder-proto": {
  "version": "1.0.1",
```

```
"resolved": "https://registry.npmjs.org/dunder-proto/-/dunder-proto-1.0.1.tgz",
"integrity": "sha512-KIN/nDJBQRcXw0MLVhZE9iQHmG68qAVIBg9CqmUYjmQlHgij9U5MFvrqkUL5FbtyyzZuOeOt0zdeRe4U
"dependencies": {
  "call-bind-apply-helpers": "^1.0.1",
  "es-errors": "^1.3.0",
  "gopd": "^1.2.0"
},
"engines": {
  "node": ">= 0.4"
}
},
"node_modules/ee-first": {
  "version": "1.1.1",
  "resolved": "https://registry.npmjs.org/ee-first/-/ee-first-1.1.1.tgz",
  "integrity": "sha512-WVnIZi9Wbqz88tDc0yFvQIjXz7fEhFG7JAggDkFkSsGfFyERyvqO0AvtMv5na+gHucXl2KHA88z9R8v
},
"node_modules/encodeurl": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-2.0.0.tgz",
  "integrity": "sha512-Q09Q0419100w18Q39e78vaNk551y2yK2U7L3wJJd/3aVb6n0zsvtF8eGeJdK76e6P+OvVG2D/G64e
"engines": {
  "node": ">= 0.8"
}
},
"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": "sha512-0w29H6xzl8aVYfoDcZsQ5v3vh5wV4y2wER5y4e2rKq244/KJ26sUwHCb/KJ9M4L6IOh5WzKq2JHw4zrH
"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-UW9JpTPF+hDCXos4tFqg+4WihkeP5xhEq6krX1SfnWfND8dzK5KlD8VDj8CwsNUt9S3fu24hWJcKBjGg07s
"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-FGgH2h8zKNim9Ijj7dankFPclClK9Cp5bm+c2gQSYePhpaG5+esrLODihlorn+Pe6FGJzWhXQotPv73jTal
"dependencies": {
  "es-errors": "^1.3.0"
},
"engines": {
  "node": ">= 0.4"
}
},
"node_modules/escape-html": {
  "version": "1.0.3",

```

```
"resolved": "https://registry.npmjs.org/escape-html/-/escape-html-1.0.3.tgz",
"integrity": "sha512-NiSupZ4OeuGwr68IGleym/ksIZMJodUGOSCZ/FSnTxcrekbvqrgdUxIJOMpijaKZVjAJrWrGs/6Jy8OMuyj9C
},
"node_modules/etag": {
  "version": "1.8.1",
  "resolved": "https://registry.npmjs.org/etag/-/etag-1.8.1.tgz",
  "integrity": "sha512-aILKxh4P0vfdU1L5p4oB+TPVFQ225XtSajWJzYMp4F4kagaVtUWKyNjPUwYkP+1mzYy3fO6Z3p3l6arqCII
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/express": {
  "version": "5.1.0",
  "resolved": "https://registry.npmjs.org/express/-/express-5.1.0.tgz",
  "integrity": "sha512-DT9ck5YIRU+8GYzzU5kT3eHGA5iL+1Zd0EutOmTE9Dtk+Tvuzd23VBU+ec7HPNSTxXYO55gPV/hq4pS
  "dependencies": {
    "accepts": "^2.0.0",
    "body-parser": "^2.2.0",
    "content-disposition": "^1.0.0",
    "content-type": "^1.0.5",
    "cookie": "^0.7.1",
    "cookie-signature": "^1.2.1",
    "debug": "^4.4.0",
    "encodeurl": "^2.0.0",
    "escape-html": "^1.0.3",
    "etag": "^1.8.1",
    "finalhandler": "^2.1.0",
    "fresh": "^2.0.0",
    "http-errors": "^2.0.0",
    "merge-descriptors": "^2.0.0",
    "mime-types": "^3.0.0",
    "on-finished": "^2.4.1",
    "once": "^1.4.0",
    "parseurl": "^1.3.3",
    "proxy-addr": "^2.0.7",
    "qs": "^6.14.0",
    "range-parser": "^1.2.1",
    "router": "^2.2.0",
    "send": "^1.1.0",
    "serve-static": "^2.2.0",
    "statuses": "^2.0.1",
    "type-is": "^2.0.1",
    "vary": "^1.1.2"
  },
  "engines": {
    "node": ">= 18"
  },
  "funding": {
    "type": "opencollective",
    "url": "https://opencollective.com/express"
  },
  "node_modules/finalhandler": {
```

```
"version": "2.1.0",
"resolved": "https://registry.npmjs.org/finalhandler/-/finalhandler-2.1.0.tgz",
"integrity": "sha512-t88Ty3d5JWQbWYgaOGCCYfXRwV1+be02WqYyIL6h0IEiUAMPm8o8qKGO01YIkOHZka2up08wvgYD
"dependencies": {
  "debug": "^4.4.0",
  "encodeurl": "^2.0.0",
  "escape-html": "^1.0.3",
  "on-finished": "^2.4.1",
  "parseurl": "^1.3.3",
  "statuses": "^2.0.1"
},
"engines": {
  "node": ">= 0.8"
}
},
"node_modules/forwarded": {
  "version": "0.2.0",
  "resolved": "https://registry.npmjs.org/forwarded/-/forwarded-0.2.0.tgz",
  "integrity": "sha512-buRG0fpBtRHSTCOASe6hD258tEubFoRLb4ZNA6NxMVHNw2gOcwHo9wyablzMzOA5z9xA9L1KNjk/Nt
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/fresh": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/fresh/-/fresh-2.0.0.tgz",
  "integrity": "sha512-Rx/WycZ60HOaqLKAI6cHRKKI7zxWbJ31MhntmtwMoaTeF7XFH9hhBp8vITaMidfIjRQ6eYWCKkaTK+yk
  "engines": {
    "node": ">= 0.8"
  }
},
"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+oVO0F744iUgR82
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/get-intrinsic": {
  "version": "1.3.0",
  "resolved": "https://registry.npmjs.org/get-intrinsic/-/get-intrinsic-1.3.0.tgz",
  "integrity": "sha512-9fSjSaos/fRIVlp+xSJIE6lfwhES7LNtKaCBlamHsjr2na1BiABJPo0mOjjz8GJDURarmCPGqaiVg5mfjb98C
  "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-sTSfBjoXBp89JvIKlefqw7U2CCebse74kiY6awiGogKtoSGbgjYE/G/+I9sF3MWFPNc9IcoOC4ODfKHfxFm",
  "dependencies": {
    "dunder-proto": "^1.0.1",
    "es-object-atoms": "^1.0.0"
  },
  "engines": {
    "node": ">= 0.4"
  }
},
"node_modules/gopd": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/gopd/-/gopd-1.2.0.tgz",
  "integrity": "sha512-ZUKRh6/kUFoAiTAiTYPZJ3hw9wNxx+BIBOijnIG9PnrJsCcSjs1wyyD6vJpaYtgnzDrKYRSqf3OO6Rfa93x",
  "engines": {
    "node": ">= 0.4"
  },
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/has-symbols": {
  "version": "1.1.0",
  "resolved": "https://registry.npmjs.org/has-symbols/-/has-symbols-1.1.0.tgz",
  "integrity": "sha512-1cDNdwJ2Jaohmb3sg4OmKaMBwuC48sYni5HUw2DvsC8LjGTLK9h+eb1X6RyuOHe4hT0ULCW68iom",
  "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": "sha512-0hJU9SCPvmMzIBdZFqNPXWa6dqh7WdH0clI9y+CyS8rG3nL48Bclra9HmKhVVUHyPWNH5Y7xDwAB7",
  "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/euBzaEjYeRqOgly7G0qviiXoJWnvEH2Z1pIBdXgbyjv34pHTSb9zoeHMyDy33+DWy5Wt9Wo+TUR
  "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/iconv-lite": {
  "version": "0.6.3",
  "resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.6.3.tgz",
  "integrity": "sha512-4fCk79wshMdZMp2rH06qWrJE4iolqLhCUH+OiulGU++RB0+94NIDL81atO7GX55uUKueo0txHNTvEyl6D7
  "dependencies": {
    "safer-buffer": ">= 2.1.2 < 3.0.0"
  },
  "engines": {
    "node": ">=0.10.0"
  }
},
"node_modules/inherits": {
  "version": "2.0.4",
  "resolved": "https://registry.npmjs.org/inherits/-/inherits-2.0.4.tgz",
  "integrity": "sha512-k/vGaX4/Yla3WzyMCvTQOXYeIHvqOKtnqBduzTHpzpQZzAskKhZ2K+EnBiSM9zGSoIFeMpXKxa4dYe
},
"node_modules/ipaddr.js": {
  "version": "1.9.1",
  "resolved": "https://registry.npmjs.org/ipaddr.js/-/ipaddr.js-1.9.1.tgz",
  "integrity": "sha512-0KI/607xoxSToH7GjN1FfSbLoU0+btTicjsQSWQlh/hZykN8KpmMf7uYwPW3R+akZ6R/w18ZIXSHBYXiU
  "engines": {
    "node": ">= 0.10"
  }
},
"node_modules/is-promise": {
  "version": "4.0.0",
  "resolved": "https://registry.npmjs.org/is-promise/-/is-promise-4.0.0.tgz",
  "integrity": "sha512-hvpol6korhJMnej285dSg6nu1+e6uxs7zG3BYAm5byqDsgJNWwxzM6z6iZiAgQR4TJ30JmBTowqZUw3V
},
"node_modules/math-intrinsics": {
  "version": "1.1.0",
  "resolved": "https://registry.npmjs.org/math-intrinsics/-/math-intrinsics-1.1.0.tgz",
  "integrity": "sha512-lXtbwEk5HTPyEwyKX6hGkYXxM9nbj64B+ilVJnC/R6B0pH5G4V3b0pVbL7DBj4tkhBApqbQUIf6F6XI9LH
  "engines": {
    "node": ">= 0.4"
  }
},

```

```
"node_modules/media-typer": {
  "version": "1.1.0",
  "resolved": "https://registry.npmjs.org/media-typer/-/media-typer-1.1.0.tgz",
  "integrity": "sha512-aisnrDP4GNe06UcKFnV5bfMNPBUw4jsLGaWwWfnH3v02GnBuXX2MCVn5RbrWo0j3pczUilYblq7fQ7Nw",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/merge-descriptors": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/merge-descriptors/-/merge-descriptors-2.0.0.tgz",
  "integrity": "sha512-Snk314V5ayFLhp3fkUREub6WtjBfPdCPY1Ln8/8munuLuiYhsABgBVWsozAG+MWMbVEvcdcpbi9R7ww",
  "engines": {
    "node": ">=18"
  },
  "funding": {
    "url": "https://github.com/sponsors/sindresorhus"
  }
},
"node_modules/mime-db": {
  "version": "1.54.0",
  "resolved": "https://registry.npmjs.org/mime-db/-/mime-db-1.54.0.tgz",
  "integrity": "sha512-aU5EJuIN2WDemCcAp2vFBfp/m4EAhWJnUNSSw0ixs7/kXbd6Pg64EmwJkNdFhB8aWt1sH2CTXrLxo/i",
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/mime-types": {
  "version": "3.0.1",
  "resolved": "https://registry.npmjs.org/mime-types/-/mime-types-3.0.1.tgz",
  "integrity": "sha512-xRc4oEhT6eaBpU1XF7AjpOFD+xQmXNB5OVKwp4tqCuBpHLS/ZbBDrc07mYTDqVMg6PfxUjjNp85O6C",
  "dependencies": {
    "mime-db": "^1.54.0"
  },
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/ms": {
  "version": "2.1.3",
  "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
  "integrity": "sha512-6Fb1496F843e6118093c0e0f266574e6b8f8c0c0",
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/negotiator": {
  "version": "1.0.0",
  "resolved": "https://registry.npmjs.org/negotiator/-/negotiator-1.0.0.tgz",
  "integrity": "sha512-8o03h1Yi2GhVU4y5ckq7uR3+SbXxXDn14R12Z9jX60GZ0UO57tPn1s3U0QVdVQO+qyYH+Qsa1ToRQ",
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/object-assign": {
  "version": "4.1.1",
```

```
"resolved": "https://registry.npmjs.org/object-assign/-/object-assign-4.1.1.tgz",
"integrity": "sha512-rJgTQnkUnH1sFw8yT6VSU3zD3sWmu6sZhIseY8VX+GRu3P6F7Fu+JNDoXfklElbLJShc3FUQHVe4cU5
"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-W67I4J2EXEGTbfHHCfrrjDfitvLANg0UIX3wFUUSTx92KXRFegMHUVgSqE+wwhAbi4WqjGg9czysTV2E
  "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-oVHzU1u1B4HW1PbF1HhK714QH36p666pht21JnK5vz1V1VxVQV3oDrOg9H2587LJ1k0K4u0j4D1S
  "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": "sha512-1k3iRkHwMu6QV22zRO8i3LFeP9K6W3iIRu4Nq2RzWw2ZSKjGOv+wM5aL1+hJBn564F0SRzygbyadUv
  "dependencies": {
    "wrappy": "1"
  }
},
"node_modules/parseurl": {
  "version": "1.3.3",
  "resolved": "https://registry.npmjs.org/parseurl/-/parseurl-1.3.3.tgz",
  "integrity": "sha512-CiyeOxFT/JZyN5m0z9PfXw4SCBJ6Sygz1Dpl0wqjlhDEGGBP1GnsUVEL0p63hoG1fcj3fHynXi9NYO4nW
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/path-to-regexp": {
  "version": "8.2.0",
  "resolved": "https://registry.npmjs.org/path-to-regexp/-/path-to-regexp-8.2.0.tgz",
  "integrity": "sha512-TdrF7fw9Rphjq4RjrW0Kp2AW0Ahwu9sRG TkS6bvDi0SCwZIEZYmcfDbEsTz8RVk0EHIS/Vd1bv3JhG+1
  "engines": {
    "node": ">=16"
  }
},
"node_modules/proxy-addr": {
```

```
"version": "2.0.7",
"resolved": "https://registry.npmjs.org/proxy-addr/-/proxy-addr-2.0.7.tgz",
"integrity": "sha512-IIQsMLSUDUPT44jdrU/O37qlnifitDP+ZwrmmZcoSKyLKvtZxpyV0n2/bD/N4tBAAZ/gJEdZU7KMraoK1+XY",
"dependencies": {
  "forwarded": "0.2.0",
  "ipaddr.js": "1.9.1"
},
"engines": {
  "node": ">= 0.10"
}
},
"node_modules/qs": {
  "version": "6.14.0",
  "resolved": "https://registry.npmjs.org/qs/-/qs-6.14.0.tgz",
  "integrity": "sha512-1Q21APplU3Yg6gIh5M0jW2VqoYUMr4drX5F8qo5+O0PjXk8vq3U7W0H4WZYT1S6QF1dVgH8r0H8H8",
  "dependencies": {
    "side-channel": "^1.1.0"
  },
  "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": "sha512-08y3C008Y5mo+Qpp8FYy7x059jjZBQjKXz0qizPitXpRL39YM7u6Oe3F2TU6K9jXZi5UaG8WU8uIUQ",
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/raw-body": {
  "version": "3.0.0",
  "resolved": "https://registry.npmjs.org/raw-body/-/raw-body-3.0.0.tgz",
  "integrity": "sha512-8YyV/bQoIQFjwLH6BfIUoVej7+0OYLKwG8bKi7DbzSgMwIqBVeQ1jZqvQsODG8uKd6Q6Mck7vDncWw",
  "dependencies": {
    "bytes": "3.1.2",
    "http-errors": "2.0.0",
    "iconv-lite": "0.6.3",
    "unpipe": "1.0.0"
  },
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/router": {
  "version": "2.2.0",
  "resolved": "https://registry.npmjs.org/router/-/router-2.2.0.tgz",
  "integrity": "sha512-2XoR3Y1YnVWQdK4nZljvE3DgNC6DZu7z3Zg3QY3Hm4Q6HlYXtJdHzL79ZJ0JmKt2X6scJmIv7RqA",
  "dependencies": {
    "debug": "^4.4.0",

```

```

    "depd": "^2.0.0",
    "is-promise": "^4.0.0",
    "parseurl": "^1.3.3",
    "path-to-regexp": "^8.0.0"
  },
  "engines": {
    "node": ">= 18"
  }
},
"node_modules/safe-buffer": {
  "version": "5.2.1",
  "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.2.1.tgz",
  "integrity": "sha512-rp3So07KcdmmKbGvgaNxQJsr7bGVSVk5S9Eq1F+ppbRo70+YeaDxkw5Dd8NPN+GD6bjnYm2VuPuCX",
  "funding": [
    {
      "type": "github",
      "url": "https://github.com/sponsors/feross"
    },
    {
      "type": "patreon",
      "url": "https://www.patreon.com/feross"
    },
    {
      "type": "consulting",
      "url": "https://feross.org/support"
    }
  ]
},
"node_modules/safer-buffer": {
  "version": "2.1.2",
  "resolved": "https://registry.npmjs.org/safer-buffer/-/safer-buffer-2.1.2.tgz",
  "integrity": "sha512-YZo3K82SD7Riyi0E1EQPojLz7kpepnSQ19IyPbHHg1XXXevb5dJI7tpyN2ADxGcQbHG7vcyRHk0cbwqcQ",
  "dependencies": {}
},
"node_modules/send": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/send/-/send-1.2.0.tgz",
  "integrity": "sha512-uaW0WwXKpL9bIXE2o0bRhoL2EGXlRZxQ2ZQ4mgcfoBxdFmQold+qWsD2jLrfZ0trjKL6vOwOj//eAwcALF",
  "dependencies": {
    "debug": "^4.3.5",
    "encodeurl": "^2.0.0",
    "escape-html": "^1.0.3",
    "etag": "^1.8.1",
    "fresh": "^2.0.0",
    "http-errors": "^2.0.0",
    "mime-types": "^3.0.1",
    "ms": "^2.1.3",
    "on-finished": "^2.4.1",
    "range-parser": "^1.2.1",
    "statuses": "^2.0.1"
  },
  "engines": {
    "node": ">= 18"
  }
}

```

```
},
"node_modules/serve-static": {
  "version": "2.2.0",
  "resolved": "https://registry.npmjs.org/serve-static/-/serve-static-2.2.0.tgz",
  "integrity": "sha512-61g9pCh0Vnh7lutZjtLGGpTA355+OPn2TyDv/6ivP2h/AdAVX9azsoxmg2/M6nZeQZNYBEwIcsne1mJd90",
  "dependencies": {
    "encodeurl": "^2.0.0",
    "escape-html": "^1.0.3",
    "parseurl": "^1.3.3",
    "send": "^1.2.0"
  },
  "engines": {
    "node": ">= 18"
  }
},
"node_modules/setprototypeof": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/setprototypeof/-/setprototypeof-1.2.0.tgz",
  "integrity": "sha512-E5LDX7Wrp85Kil5bhZv46j8jOeboKq5JMmYM3gVGdGH8xFpPWXUMsNrIODCrkoxMEeNi/XZlwRvY4X",
},
"node_modules/side-channel": {
  "version": "1.1.0",
  "resolved": "https://registry.npmjs.org/side-channel/-/side-channel-1.1.0.tgz",
  "integrity": "sha512-ZX99e6tRweoUXqR+VBrsIhda51Nh5MTQwou5tnUDgbtyM0dBgmhEDtWGP/xbKn6hqfPRHujUNwz5fy/w",
  "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+wzCmDEkc9y7NsYxeF4u7Btsn1ZuwgwJGxImImHicJArlP4R0yX4c2K",
  "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-VCjCNfgMsby3tTdo02nbjtM/ewra6jPHmpThenkTYh8pG9ucZ/1P8So4u4FGBek/BjpOVsDCMoLA/iuBKIF",
  "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/tlpvldyrnem4aeQp4hai3gythswg6p01oSoTI58rcpiFAjF2br2Ak2",
  "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/7PrK06rYLIzFMlAF+773iwpzsqRIFgbMLbTcLD6cOao82TaWefPXQvB2fOC4AjuYSEndS7N/mT",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/toidentifier": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/toidentifier/-/toidentifier-1.0.1.tgz",
  "integrity": "sha512-o5sSPKEkg/DIQNmH43V0/uerLrpzVedkUh8tGNvaeXpfpuwjKenISox/2O/BTIZUtEe+JG7s5YhEz608PIA",
  "engines": {
    "node": ">=0.6"
  }
},
"node_modules/type-is": {
  "version": "2.0.1",
```



```
"resolved": "https://registry.npmjs.org/type-is/-/type-is-2.0.1.tgz",
"integrity": "sha512-OZs6gsjF4vMp32qrCbiVSkrFmXtG/AZhY3t0iAMrMBiAZyV9oALtXO8hsrHbMXF9x6L3grIFuwW2oAz7cav
"dependencies": {
  "content-type": "^1.0.5",
  "media-typer": "^1.1.0",
  "mime-types": "^3.0.0"
},
"engines": {
  "node": ">= 0.6"
}
},
"node_modules/unpipe": {
  "version": "1.0.0",
  "resolved": "https://registry.npmjs.org/unpipe/-/unpipe-1.0.0.tgz",
  "integrity": "sha512-pjy2bYhSsufwWIKwPc+l3cN7+wuJIK6uz0YdJEOIQDbI6jo/YIPi4mb8agUkVC8BF7V8NuzeyPNqRksA3hz
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/uuid": {
  "version": "11.1.0",
  "resolved": "https://registry.npmjs.org/uuid/-/uuid-11.1.0.tgz",
  "integrity": "sha512-0/A9rDy9P7cJ+8w1c9WD9V//9Wj15Ce2MPz8Ri6032usz+NfePxx5AcN3bN+r6ZL6jEo066/yNYB3tn4pQE
  "funding": [
    "https://github.com/sponsors/broofa",
    "https://github.com/sponsors/ctavan"
  ],
  "bin": {
    "uuid": "dist/esm/bin/uuid"
  }
},
"node_modules/vary": {
  "version": "1.1.2",
  "resolved": "https://registry.npmjs.org/vary/-/vary-1.1.2.tgz",
  "integrity": "sha512-BNGbWLfd0eUPabhkXUVm0j8uuvREyTh5ovRa/dyow/BqAbZJyC+5fU+lzQOzmAKzYqYRAISoRhdQr3e
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/wrappy": {
  "version": "1.0.2",
  "resolved": "https://registry.npmjs.org/wrappy/-/wrappy-1.0.2.tgz",
  "integrity": "sha512-3ZcVOH1weKjaTZpjqwO14m9d1Yf8RKtaYt45LW/ZKZtE5dKZ+pJ/Z5klpKu1d8d8JuvRkIs4fNqB6wYoCYv
}
}
}
```

File: relatorio-frequencias-aluno.html

```
=====

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Relatório de Frequências por Aluno - Controle de Frequência</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
  <link rel="stylesheet" href="/css/style.css">
</head>
<body>
  <div class="container py-4">
    <h1 class="mb-5 text-center">Relatório de Frequências por Aluno</h1>
    <div class="card">
      <div class="card-header">Relatório de Frequências por Aluno</div>
      <div class="card-body">
        <form id="formRelatorioAluno">
          <div class="mb-3">
            <label for="alunoRelatorio" class="form-label">Aluno</label>
            <select class="form-select" id="alunoRelatorio" required>
              <option value="" disabled selected>Selecione um aluno</option>
            </select>
          </div>
          <button type="submit" class="btn btn-primary w-100" id="btnConsultarRelatorioAluno">Consultar</button>
        </form>
        <div id="relatorioAluno" class="mt-3"></div>
      </div>
    </div>
    <div class="mt-4 text-center">
      <a href="/index.html" class="btn btn-secondary">Voltar ao Menu</a>
    </div>
  </div>
  <!-- Scripts -->
  <script src="/js/model.js"></script>
  <script src="/js/controller.js"></script>
  <script src="/js/view.js"></script>
</body>
</html>
```

File: data\alunos.json

=====

```
[
  {
    "id": "a37baf30-79e7-4035-8848-54be37311b2c",
    "nome": "Luna",
    "turmaId": "91353437-ebfa-48f1-a3d1-cc64912402ec",
    "dataCadastro": "2025-04-06T22:29:00.368Z"
  },
  {
    "id": "df0622ff-b5b2-48ae-9610-510cf5429749",
    "nome": "Arink",
    "turmaId": "c76d2448-f350-466b-9c44-d65d4d75ba53",
    "dataCadastro": "2025-04-06T22:53:04.642Z"
  },
  {
    "id": "15e860cb-1a8c-4252-95a2-b68214d043ca",
    "nome": "Lupita",
    "turmaId": "2332e1b9-5941-4fc3-8618-d7b940eddb7c",
    "dataCadastro": "2025-04-06T23:11:47.385Z"
  }
]
```

File: marcar-presenca.html

```
=====

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Marcar Presença - Controle de Frequência</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
  <link rel="stylesheet" href="/css/style.css">
</head>
<body>
  <div class="container py-4">
    <h1 class="mb-5 text-center">Marcar Presença</h1>
    <div class="card">
      <div class="card-header">Marcar Presença</div>
      <div class="card-body">
        <form id="formPresenca">
          <div class="row">
            <div class="col-md-6 mb-3">
              <label for="turmaPresenca" class="form-label">Turma</label>
              <select class="form-select" id="turmaPresenca" required>
                <option value="" disabled selected>Selecione uma turma</option>
              </select>
            </div>
            <div class="col-md-6 mb-3">
              <label for="dataPresenca" class="form-label">Data</label>
              <input type="date" class="form-control" id="dataPresenca" required max="{{hoje}}">
            </div>
          </div>
          <button type="submit" class="btn btn-primary w-100" id="btnCarregarLista">Carregar Lista</button>
        </form>
        <div id="listaPresenca" class="mt-3"></div>
      </div>
    </div>
    <div class="mt-4 text-center">
      <a href="/index.html" class="btn btn-secondary">Voltar ao Menu</a>
    </div>
  </div>
  <!-- Scripts -->
  <script src="/js/model.js"></script>
  <script src="/js/controller.js"></script>
  <script src="/js/view.js"></script>
</body>
</html>
```

File: cadastros.html

```
=====

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cadastros - Controle de Frequência</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
  <link rel="stylesheet" href="/css/style.css">
</head>
<body>
  <div class="container py-4">
    <h1 class="mb-5 text-center">Cadastros</h1>
    <div class="row g-4">
      <!-- Cadastro de Turma -->
      <div class="col-md-6">
        <div class="card">
          <div class="card-header">Cadastrar Turma</div>
          <div class="card-body">
            <form id="formTurma">
              <div class="mb-3">
                <label for="nomeTurma" class="form-label">Nome da Turma</label>
                <input type="text" class="form-control" id="nomeTurma" required aria-describedby="nomeTurmaHelp">
                <div id="nomeTurmaHelp" class="form-text">Digite o nome da turma (ex.: Turma A).</div>
              </div>
            </form>
          </div>
        </div>
      </div>
      <div class="col-md-6">
        <div class="card">
          <div class="card-header">Cadastrar Aluno</div>
          <div class="card-body">
            <form id="formAluno">
              <div class="mb-3">
                <label for="nomeAluno" class="form-label">Nome do Aluno</label>
                <input type="text" class="form-control" id="nomeAluno" required aria-describedby="nomeAlunoHelp">
                <div id="nomeAlunoHelp" class="form-text">Digite o nome do aluno (ex.: João da Silva).</div>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

```
<label for="nomeAluno" class="form-label">Nome do Aluno</label>
<input type="text" class="form-control" id="nomeAluno" required aria-describedby="nomeAlunoHelp">
<div id="nomeAlunoHelp" class="form-text">Digite o nome completo do aluno.</div>
</div>
<div class="mb-3">
  <label for="turmaAluno" class="form-label">Turma</label>
  <select class="form-select" id="turmaAluno" required>
    <option value="" disabled selected>Selecione uma turma</option>
  </select>
</div>
<button type="submit" class="btn btn-primary w-100">Cadastrar Aluno</button>
</form>
<div id="alunoFeedback" class="mt-3"></div>
</div>
</div>
</div>
</div>
<div class="mt-4 text-center">
  <a href="/index.html" class="btn btn-secondary">Voltar ao Menu</a>
</div>
</div>
<!-- Scripts -->
<script src="/js/model.js"></script>
<script src="/js/controller.js"></script>
<script src="/js/view.js"></script>
</body>
</html>
```

File: server.js

```
=====

// server.js
const express = require('express');
const cors = require('cors');
const { v4: uuidv4 } = require('uuid');
const fs = require('fs');
const path = require('path');

const app = express();
const PORT = process.env.PORT || 3000;

// Middleware
app.use(cors());
app.use(express.json());
app.use(express.static(path.join(__dirname, 'public')));

// Data file paths
const DATA_DIR = path.join(__dirname, 'data');
const TURMAS_FILE = path.join(DATA_DIR, 'turmas.json');
const ALUNOS_FILE = path.join(DATA_DIR, 'alunos.json');
const PRESENCAS_FILE = path.join(DATA_DIR, 'presencas.json');

// Ensure data directory exists
if (!fs.existsSync(DATA_DIR)) {
  fs.mkdirSync(DATA_DIR, { recursive: true });
}

// Initialize data files if they don't exist
if (!fs.existsSync(TURMAS_FILE)) {
  fs.writeFileSync(TURMAS_FILE, JSON.stringify([]));
}

if (!fs.existsSync(ALUNOS_FILE)) {
  fs.writeFileSync(ALUNOS_FILE, JSON.stringify([]));
}

if (!fs.existsSync(PRESENCAS_FILE)) {
  fs.writeFileSync(PRESENCAS_FILE, JSON.stringify([]));
}

// Helper functions for data operations
const readData = (filePath) => {
  try {
    const data = fs.readFileSync(filePath, 'utf8');
    return JSON.parse(data);
  } catch (error) {
    console.error(`Error reading ${filePath}:`, error);
    return [];
  }
};
```

```

const writeData = (filePath, data) => {
  try {
    fs.writeFileSync(filePath, JSON.stringify(data, null, 2));
    return true;
  } catch (error) {
    console.error(`Error writing to ${filePath}:`, error);
    return false;
  }
};

// API Routes for Turmas (Classes)
app.get('/api/turmas', (req, res) => {
  const turmas = readData(TURMAS_FILE);
  res.json(turmas);
});

app.post('/api/turmas', (req, res) => {
  try {
    const { nome, turno } = req.body;
    if (!nome || !turno) {
      return res.status(400).json({ error: 'Nome e turno são obrigatórios' });
    }

    const turmas = readData(TURMAS_FILE);
    const novaTurma = {
      id: uuidv4(),
      nome,
      turno,
      dataCriacao: new Date().toISOString()
    };

    turmas.push(novaTurma);
    writeData(TURMAS_FILE, turmas);
    res.status(201).json(novaTurma);
  } catch (error) {
    console.error('Error creating turma:', error);
    res.status(500).json({ error: 'Erro ao criar turma' });
  }
});

// API Routes for Alunos (Students)
app.get('/api/alunos', (req, res) => {
  const alunos = readData(ALUNOS_FILE);
  res.json(alunos);
});

app.get('/api/alunos/turma/:turmaId', (req, res) => {
  const { turmaId } = req.params;
  const alunos = readData(ALUNOS_FILE);
  const alunosDaTurma = alunos.filter(aluno => aluno.turmaId === turmaId);
  res.json(alunosDaTurma);
});

```



```

app.post('/api/alunos', (req, res) => {
  try {
    const { nome, turmalId } = req.body;
    if (!nome || !turmalId) {
      return res.status(400).json({ error: 'Nome e turmalId são obrigatórios' });
    }

    const turmas = readData(TURMAS_FILE);
    const turmaExists = turmas.some(turma => turma.id === turmalId);

    if (!turmaExists) {
      return res.status(404).json({ error: 'Turma não encontrada' });
    }

    const alunos = readData(ALUNOS_FILE);
    const novoAluno = {
      id: uuidv4(),
      nome,
      turmalId,
      dataCadastro: new Date().toISOString()
    };

    alunos.push(novoAluno);
    writeData(ALUNOS_FILE, alunos);
    res.status(201).json(novoAluno);
  } catch (error) {
    console.error('Error creating aluno:', error);
    res.status(500).json({ error: 'Erro ao criar aluno' });
  }
});

// API Routes for Presença (Attendance)
app.get('/api/presencas', (req, res) => {
  const presencas = readData(PRESENCAS_FILE);
  res.json(presencas);
});

// Get attendance records by class and date
app.get('/api/presencas/turma/:turmalId/data/:data', (req, res) => {
  try {
    const { turmalId, data } = req.params;

    // Validate parameters
    if (!turmalId || !data) {
      return res.status(400).json({ error: 'ID da turma e data são obrigatórios' });
    }

    const presencas = readData(PRESENCAS_FILE);
    const presencasFiltradas = presencas.filter(p =>
      p.turmalId === turmalId && p.data === data
    );
  }
});

```

```

    res.json(presencasFiltradas);
  } catch (error) {
    console.error('Error getting attendance by class and date:', error);
    res.status(500).json({ error: 'Erro ao buscar presenças' });
  }
});

```

// Get attendance records by student

```

app.get('/api/presencas/aluno/:alunold', (req, res) => {
  try {
    const { alunold } = req.params;

    // Validate parameters
    if (!alunold) {
      return res.status(400).json({ error: 'ID do aluno é obrigatório' });
    }

    const presencas = readData(PRESENCAS_FILE);
    const registrosPresenca = presencas.filter(p => {
      return p.registros.some(r => r.alunold === alunold);
    });

```

// Format the data to be more usable on the frontend

```

const historicoAluno = registrosPresenca.map(p => {
  const registro = p.registros.find(r => r.alunold === alunold);
  return {
    data: p.data,
    turmaid: p.turmaid,
    presente: registro.presente
  };
});

res.json(historicoAluno);
} catch (error) {
  console.error('Error getting student attendance:', error);
  res.status(500).json({ error: 'Erro ao buscar presença do aluno' });
}
});

```

// Register attendance for a class on a specific date

```

app.post('/api/presencas', (req, res) => {
  try {
    const { turmaid, data, registros } = req.body;

    // Validate request
    if (!turmaid || !data || !registros || !Array.isArray(registros)) {
      return res.status(400).json({
        error: 'Dados incompletos. Forneça turmaid, data e um array de registros'
      });
    }

```

// Validate each record has alunold and presente status

```

for (const registro of registros) {

```

```

    if (!registro.alunold || typeof registro.presente !== 'boolean') {
      return res.status(400).json({
        error: 'Cada registro deve ter alunold e status de presença (boolean)'
      });
    }
  }
}

const presencas = readData(PRESENCAS_FILE);

// Check if there's already a record for this class and date
const existingIndex = presencas.findIndex(p =>
  p.turmalid === turmalid && p.data === data
);

if (existingIndex >= 0) {
  // Update existing record
  presencas[existingIndex].registros = registros;
  presencas[existingIndex].updatedAt = new Date().toISOString();
} else {
  // Create new record
  const novoRegistro = {
    id: uuidv4(),
    turmalid,
    data,
    registros,
    createdAt: new Date().toISOString(),
    updatedAt: new Date().toISOString()
  };
  presencas.push(novoRegistro);
}

writeData(PRESENCAS_FILE, presencas);

res.status(201).json({
  success: true,
  message: 'Presenças registradas com sucesso'
});
} catch (error) {
  console.error('Error registering attendance:', error);
  res.status(500).json({ error: 'Erro ao registrar presenças' });
}
});

// Start the server
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});

```

File: js\controller.js

```
=====

const Controller = {
  init: async function() {
    document.getElementById('formTurma').addEventListener('submit', this.handleAddTurma);
    document.getElementById('formAluno').addEventListener('submit', this.handleAddAluno);
    await this.loadTurmas();
    await View.atualizarSeletorTurmas('turmaPresenca');
    await View.atualizarSeletorTurmas('turmaFrequencia');
    await View.atualizarSeletorAlunos('alunoRelatorio'); // Carregar alunos no seletor de relatório
  },

  loadTurmas: async function () {
    try {
      await View.atualizarListaTurmas();
    } catch (error) {
      console.error('Erro ao carregar turmas:', error);
      alert('Não foi possível carregar as turmas. Verifique se o servidor está rodando.');
```

```

    } else {
      alert('Erro ao cadastrar aluno. Tente novamente.');
```

```
    }
```

```

  } catch (error) {
    console.error('Erro ao adicionar aluno:', error);
    alert('Erro ao cadastrar aluno. Verifique o console para mais detalhes.');
```

```
  }
```

```
},
```

```

handleLoadAttendanceList: async function (event) {
  event.preventDefault(); // Adicionado para evitar recarregamento da página
  const turmald = document.getElementById('turmaPresenca').value;
  const data = document.getElementById('dataPresenca').value;
```

```

  try {
```

```
    if (!turmald || !data) {
```

```
      alert('Por favor, selecione uma turma e uma data.');
```

```
      return;
```

```
    }
```

```
    await View.renderizarListaPresenca(turmald, data);
```

```
  } catch (error) {
```

```
    console.error('Erro ao carregar lista de presença:', error);
```

```
    alert('Erro ao carregar lista de presença. Verifique o console para mais detalhes.');
```

```
  }
```

```
},
```

```

handleShowClassReport: async function (event) {
  event.preventDefault(); // Adicionado
  const turmald = document.getElementById('turmaFrequencia').value;
  const data = document.getElementById('dataFrequencia').value;
```

```

  try {
```

```
    if (!turmald || !data) {
```

```
      alert('Por favor, selecione uma turma e uma data.');
```

```
      return;
```

```
    }
```

```
    await View.renderizarRelatorioFrequencia(turmald, data);
```

```
  } catch (error) {
```

```
    console.error('Erro ao exibir relatório de frequência:', error);
```

```
    alert('Erro ao exibir relatório de frequência. Verifique o console para mais detalhes.');
```

```
  }
```

```
},
```

```

handleShowStudentReport: async function (event) {
  event.preventDefault(); // Adicionado
  const alunold = document.getElementById('alunoRelatorio').value;
```

```

  try {
```

```
    if (!alunold) {
```

```
      alert('Por favor, selecione um aluno.');
```

```
      return;
```

```
    }
```

```
    await View.renderizarRelatorioAluno(alunold);
```

```
    } catch (error) {  
        console.error('Erro ao exibir relatório do aluno:', error);  
        alert('Erro ao exibir relatório do aluno. Verifique o console para mais detalhes.');
```

```
    }  
};  
  
document.getElementById('dataPresenca').max = new Date().toISOString().split('T')[0];
```

```
// Initialize the controller when the DOM is loaded  
document.addEventListener('DOMContentLoaded', function () {  
    Controller.init();  
});
```

File: README.md

Controle de Frequência de Alunos

Este projeto é uma atividade da disciplina de Arquitetura de Software da UNIRV, no curso de Engenharia de Software.

Objetivo

Desenvolver um sistema para controle de presença de alunos, com as seguintes funcionalidades:

- Cadastro de turmas
- Cadastro de alunos (associados a uma turma)
- Marcação de presença por data (a ser implementado)
- Consulta de frequência (a ser implementado)
- Relatório por aluno (a ser implementado)

Arquitetura

O sistema utiliza uma **Arquitetura em Camadas (MVC)** com Backend:

- **Model**: Gerencia dados e interage com o backend via API
- **View**: HTML com Bootstrap para uma interface simples e responsiva
- **Controller**: Conecta eventos da interface com as ações do Model
- **Backend**: API RESTful com Express.js e persistência em arquivos JSON

Decisões de Projeto

- **HTML, CSS e JS**: para interface simples e responsiva
- **Bootstrap**: para uma interface com mínima codificação CSS
- **Express.js**: framework backend leve e eficiente
- **API REST**: para separação clara entre frontend e backend
- **Persistência em arquivos JSON**: solução simples sem necessidade de configurar banco de dados

Tecnologias Utilizadas

- **Frontend**: HTML, CSS, JavaScript (vanilla)
- **CSS Framework**: Bootstrap 5
- **Backend**: Node.js, Express.js
- **Persistência**: Arquivos JSON
- **Ferramentas**: UUID para geração de IDs, CORS para comunicação cross-origin

Como usar

1. Instale o Node.js (<https://nodejs.org>)
2. Clone o repositório
3. Instale as dependências:

...

```
npm install
```

...

4. Inicie o servidor:

...

```
npm start
```

...

5. Acesse o sistema em `http://localhost:3000`

Estrutura do Projeto

...

```
controle-frequencia/  
  data/          # Armazenamento de dados (JSON)  
    alunos.json  
    turmas.json  
  public/        # Frontend  
    css/  
      style.css  
    js/  
      model.js  
      view.js  
      controller.js  
    index.html  
  server.js      # Servidor Express  
  package.json   # Configuração do projeto  
  README.md
```

...

Desenvolvido por Marques Vinícius Melo Martins

File: index.html

```
=====

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Controle de Frequência - Menu</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
  <link rel="stylesheet" href="/css/style.css">
</head>
<body>
  <div class="container py-5">
    <h1 class="text-center mb-5">Controle de Frequência</h1>
    <div class="row justify-content-center">
      <div class="col-md-8">
        <div class="d-grid gap-3">
          <a href="/cadastros.html" class="btn btn-primary btn-lg">Cadastros</a>
          <a href="/marcar-presenca.html" class="btn btn-primary btn-lg">Marcar Presença</a>
          <a href="/consultar-frequencias-turma.html" class="btn btn-primary btn-lg">Consultar Frequências por Turma</a>
          <a href="/relatorio-frequencias-aluno.html" class="btn btn-primary btn-lg">Relatório de Frequências por Aluno</a>
        </div>
      </div>
    </div>
  </div>
  <!-- Scripts -->
  <script src="/js/model.js"></script>
  <script src="/js/controller.js"></script>
  <script src="/js/view.js"></script>
</body>
</html>
```

File: js\view.js

```
=====

const View = {

  // Atualiza um seletor genérico de turmas (usado em presença e consulta)
  atualizarSeletorTurmas: async function (seletorId = 'turmaAluno'){
    try {
      const select = document.getElementById(seletorId);
      if (!select) return;

      select.innerHTML = "";

      const defaultOption = document.createElement("option");
      defaultOption.value = "";
      defaultOption.textContent = "Selecione uma turma";
      defaultOption.disabled = true;
      defaultOption.selected = true;
      select.appendChild(defaultOption);

      const turmas = await Model.getTurmas();

      if (turmas.length === 0) {
        const option = document.createElement("option");
        option.value = "";
        option.textContent = "Nenhuma turma cadastrada";
        option.disabled = true;
        select.appendChild(option);
      } else {
        turmas.forEach((turma) => {
          const option = document.createElement("option");
          option.value = turma.id;
          option.textContent = `${turma.nome} (${turma.turno})`;
          select.appendChild(option);
        });
      }
    } catch (error) {
      console.error("Erro ao atualizar seletor de turmas:", error);
    }
  },

  // Atualiza o seletor de alunos (usado no relatório por aluno)
  atualizarSeletorAlunos: async function (seletorId) {
    try {
      const select = document.getElementById(seletorId);
      if (!select) return;

      select.innerHTML = "";

      const defaultOption = document.createElement("option");
      defaultOption.value = "";
      defaultOption.textContent = "Selecione um aluno";
```

```

defaultOption.disabled = true;
defaultOption.selected = true;
select.appendChild(defaultOption);

const alunos = await Model.getAlunos();

if (alunos.length === 0) {
  const option = document.createElement("option");
  option.value = "";
  option.textContent = "Nenhum aluno cadastrado";
  option.disabled = true;
  select.appendChild(option);
} else {
  alunos.sort((a, b) => a.nome.localeCompare(b.nome));
  alunos.forEach((aluno) => {
    const option = document.createElement("option");
    option.value = aluno.id;
    option.textContent = aluno.nome;
    select.appendChild(option);
  });
}
} catch (error) {
  console.error("Erro ao atualizar seletor de alunos:", error);
}
},

// Renderiza a lista de presença para marcação
renderizarListaPresenca: async function (turmaId, data) {
  try {
    const container = document.getElementById("listaPresenca");
    if (!container) return;
    container.innerHTML = "<p>Carregando lista de alunos...</p>";

    const alunos = await Model.getAlunosByTurma(turmaId);
    if (alunos.length === 0) {
      container.innerHTML = "<p>Nenhum aluno cadastrado para esta turma.</p>";
      return;
    }

    const presencas = await Model.getPresencasByTurmaData(turmaId, data);
    let registrosExistentes = [];
    if (presencas.length > 0) {
      registrosExistentes = presencas[0].registros;
    }

    const form = document.createElement("form");
    form.id = "formPresencaSubmit";
    form.classList.add("mb-4");

    const table = document.createElement("table");
    table.classList.add("table", "table-striped");

    const thead = document.createElement("thead");

```

```

const headerRow = document.createElement("tr");
const thNome = document.createElement("th");
thNome.textContent = "Nome";
const thPresenca = document.createElement("th");
thPresenca.textContent = "Presente";
headerRow.appendChild(thNome);
headerRow.appendChild(thPresenca);
thead.appendChild(headerRow);
table.appendChild(thead);

const tbody = document.createElement("tbody");
alunos.forEach((aluno) => {
  const row = document.createElement("tr");
  const tdNome = document.createElement("td");
  tdNome.textContent = aluno.nome;
  const tdPresenca = document.createElement("td");
  const checkPresente = document.createElement("input");
  checkPresente.type = "checkbox";
  checkPresente.name = `aluno_${aluno.id}`;
  checkPresente.value = "presente";
  checkPresente.classList.add("form-check-input");
  checkPresente.dataset.alunoid = aluno.id;

  const registroExistente = registrosExistentes.find((r) => r.alunoid === aluno.id);
  if (registroExistente && registroExistente.presente) {
    checkPresente.checked = true;
  }

  tdPresenca.appendChild(checkPresente);
  row.appendChild(tdNome);
  row.appendChild(tdPresenca);
  tbody.appendChild(row);
});

table.appendChild(tbody);
form.appendChild(table);

const submitBtn = document.createElement("button");
submitBtn.type = "submit";
submitBtn.classList.add("btn", "btn-primary");
submitBtn.textContent = "Salvar Presenças";
form.appendChild(submitBtn);

container.innerHTML = "";
container.appendChild(form);

form.addEventListener("submit", async (event) => {
  event.preventDefault();
  const registros = [];
  alunos.forEach((aluno) => {
    const checkbox = document.querySelector(`input[data-aluno-id="${aluno.id}"]`);
    registros.push({
      alunoid: aluno.id,

```

```

        presente: checkbox.checked,
    });
});
try {
    await Model.registrarPresencas(turmalId, data, registros);
    alert("Presenças registradas com sucesso!");
    await this.renderizarListaPresenca(turmalId, data); // Atualiza a lista após salvar
} catch (error) {
    console.error("Erro ao salvar presenças:", error);
    alert("Erro ao salvar presenças. Verifique o console.");
}
});
} catch (error) {
    console.error("Erro ao renderizar lista de presença:", error);
    container.innerHTML = "<p>Erro ao carregar lista. Verifique o console.</p>";
}
},

```

// Renderiza o relatório de frequência por turma e data

renderizarRelatorioFrequencia: async function (turmalId, data) {

```

    try {
        const container = document.getElementById("relatorioFrequencia");
        if (!container) return;

        container.innerHTML = "<p>Carregando relatório de frequência...</p>";

```

```

        const turmas = await Model.getTurmas();
        const turma = turmas.find((t) => t.id === turmalId);

```

```

        if (!turma) {
            container.innerHTML = "<p>Turma não encontrada.</p>";
            return;
        }

```

```

        const presencas = await Model.getPresencasByTurmaData(turmalId, data);

```

```

        if (presencas.length === 0) {
            container.innerHTML = `<div class="alert alert-info">Não há registro de frequência para a turma ${turma.nome} na data ${data}</div>`;
            return;
        }

```

```

        const alunos = await Model.getAlunosByTurma(turmalId);

```

```

        if (alunos.length === 0) {
            container.innerHTML = "<p>Nenhum aluno cadastrado para esta turma.</p>";
            return;
        }

```

```

        const reportHeader = document.createElement("div");
        reportHeader.innerHTML = `
            <h3>Relatório de Frequência</h3>
            <p>Turma: ${turma.nome} (${turma.turno})</p>
            <p>Data: ${data}</p>

```

```
`;
```

```
const table = document.createElement("table");
table.classList.add("table", "table-striped");
```

```
const thead = document.createElement("thead");
const headerRow = document.createElement("tr");
const thNome = document.createElement("th");
thNome.textContent = "Nome";
const thStatus = document.createElement("th");
thStatus.textContent = "Status";
headerRow.appendChild(thNome);
headerRow.appendChild(thStatus);
thead.appendChild(headerRow);
table.appendChild(thead);
```

```
const tbody = document.createElement("tbody");
const registros = presencas[0].registros;
```

```
let presentes = 0;
let ausentes = 0;
```

```
alunos.forEach((aluno) => {
  const row = document.createElement("tr");
  const tdNome = document.createElement("td");
  tdNome.textContent = aluno.nome;
  const tdStatus = document.createElement("td");

  const registro = registros.find((r) => r.alunoid === aluno.id);
  if (registro && registro.presente) {
    tdStatus.textContent = "Presente";
    tdStatus.classList.add("text-success");
    presentes++;
  } else {
    tdStatus.textContent = "Ausente";
    tdStatus.classList.add("text-danger");
    ausentes++;
  }

  row.appendChild(tdNome);
  row.appendChild(tdStatus);
  tbody.appendChild(row);
});
```

```
table.appendChild(tbody);
```

```
const summary = document.createElement("div");
const total = alunos.length;
const percentPresentes = ((presentes / total) * 100).toFixed(1);
```

```
summary.innerHTML = `
<p class="mt-3">
  <strong>Resumo:</strong> ${presentes} presentes (${percentPresentes}%) e ${ausentes} ausentes de um total de ${total}
```

```

    </p>
    `;

    container.innerHTML = "";
    container.appendChild(reportHeader);
    container.appendChild(table);
    container.appendChild(summary);
  } catch (error) {
    console.error("Erro ao renderizar relatório de frequência:", error);
    const container = document.getElementById("relatorioFrequencia");
    if (container) {
      container.innerHTML = "<p>Erro ao carregar relatório. Verifique o console para mais detalhes.</p>";
    }
  }
},

// Renderiza o relatório de frequência individual por aluno
renderizarRelatorioAluno: async function (alunold) {
  try {
    const container = document.getElementById("relatorioAluno");
    if (!container) return;

    container.innerHTML = "<p>Carregando relatório do aluno...</p>";

    const alunos = await Model.getAlunos();
    const aluno = alunos.find((a) => a.id === alunold);

    if (!aluno) {
      container.innerHTML = "<p>Aluno não encontrado.</p>";
      return;
    }

    const turmas = await Model.getTurmas();
    const turma = turmas.find((t) => t.id === aluno.turmaid);

    const historicoPresenca = await Model.getPresencasByAluno(alunold);

    const reportHeader = document.createElement("div");
    reportHeader.innerHTML = `
      <h3>Relatório de Frequência Individual</h3>
      <p>Aluno: ${aluno.nome}</p>
      <p>Turma: ${turma ? `${turma.nome} (${turma.turno})` : "Não encontrada"}</p>
    `;

    if (historicoPresenca.length === 0) {
      container.innerHTML = "";
      container.appendChild(reportHeader);
      container.innerHTML += `<div class="alert alert-info">Não há registros de frequência para este aluno.</div>`;
      return;
    }

    const table = document.createElement("table");
    table.classList.add("table", "table-striped");
  }
}

```

```

const thead = document.createElement("thead");
const headerRow = document.createElement("tr");
const thData = document.createElement("th");
thData.textContent = "Data";
const thStatus = document.createElement("th");
thStatus.textContent = "Status";
headerRow.appendChild(thData);
headerRow.appendChild(thStatus);
thead.appendChild(headerRow);
table.appendChild(thead);

const tbody = document.createElement("tbody");

historicoPresenca.sort((a, b) => new Date(b.data) - new Date(a.data));

let presentes = 0;
let ausentes = 0;

historicoPresenca.forEach((registro) => {
  const row = document.createElement("tr");
  const tdData = document.createElement("td");
  tdData.textContent = registro.data;
  const tdStatus = document.createElement("td");

  if (registro.presente) {
    tdStatus.textContent = "Presente";
    tdStatus.classList.add("text-success");
    presentes++;
  } else {
    tdStatus.textContent = "Ausente";
    tdStatus.classList.add("text-danger");
    ausentes++;
  }

  row.appendChild(tdData);
  row.appendChild(tdStatus);
  tbody.appendChild(row);
});

table.appendChild(tbody);

const summary = document.createElement("div");
const total = historicoPresenca.length;
const percentPresentes = ((presentes / total) * 100).toFixed(1);

summary.innerHTML = `
<p class="mt-3">
  <strong>Resumo:</strong> Presente em ${presentes} de ${total} aulas (${percentPresentes}% de frequência).
</p>
`;

container.innerHTML = "";

```



```
    container.appendChild(reportHeader);
    container.appendChild(table);
    container.appendChild(summary);
  } catch (error) {
    console.error("Erro ao renderizar relatório do aluno:", error);
    const container = document.getElementById("relatorioAluno");
    if (container) {
      container.innerHTML = "<p>Erro ao carregar relatório. Verifique o console para mais detalhes.</p>";
    }
  }
},
};
```

File: data\turmas.json

=====

```
[
  {
    "id": "91353437-ebfa-48f1-a3d1-cc64912402ec",
    "nome": "Turma A",
    "turno": "Manhã",
    "dataCriacao": "2025-04-06T22:28:52.471Z"
  },
  {
    "id": "c76d2448-f350-466b-9c44-d65d4d75ba53",
    "nome": "Turma c",
    "turno": "Noite",
    "dataCriacao": "2025-04-06T22:52:56.344Z"
  },
  {
    "id": "2332e1b9-5941-4fc3-8618-d7b940eddb7c",
    "nome": "Turma A2",
    "turno": "Manhã",
    "dataCriacao": "2025-04-06T23:11:37.244Z"
  }
]
```

File: data\presencas.json

=====

File: js\model.js

```
=====

const Model = {
  API_URL: 'http://localhost:3000/api',

  async getTurmas() {
    try {
      const response = await fetch(`${this.API_URL}/turmas`);
      if (!response.ok) {
        throw new Error('Erro ao buscar turmas');
      }
      return await response.json();
    } catch (error) {
      console.error('Erro ao buscar turmas:', error);
      return [];
    }
  },

  async addTurma(nome, turno) {
    try {
      const response = await fetch(`${this.API_URL}/turmas`, {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({ nome, turno }),
      });

      if (!response.ok) {
        throw new Error('Erro ao criar turma');
      }

      return await response.json();
    } catch (error) {
      console.error('Erro ao adicionar turma:', error);
      return null;
    }
  },

  async getAlunos() {
    try {
      const response = await fetch(`${this.API_URL}/alunos`);
      if (!response.ok) {
        throw new Error('Erro ao buscar alunos');
      }
      return await response.json();
    } catch (error) {
      console.error('Erro ao buscar alunos:', error);
      return [];
    }
  },
}
```

```

async getAlunosByTurma(turmaid) {
  try {
    const response = await fetch(`${this.API_URL}/alunos/turma/${turmaid}`);
    if (!response.ok) {
      throw new Error('Erro ao buscar alunos da turma');
    }
    return await response.json();
  } catch (error) {
    console.error('Erro ao buscar alunos da turma:', error);
    return [];
  }
},

```

```

async addAluno(nome, turmaid) {
  try {
    const response = await fetch(`${this.API_URL}/alunos`, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({ nome, turmaid }),
    });

    if (!response.ok) {
      throw new Error('Erro ao criar aluno');
    }

    return await response.json();
  } catch (error) {
    console.error('Erro ao adicionar aluno:', error);
    return null;
  }
},

```

// Attendance Methods

```

async registrarPresencas(turmaid, data, registros) {
  try {
    const response = await fetch(`${this.API_URL}/presencas`, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({ turmaid, data, registros }),
    });

    if (!response.ok) {
      throw new Error('Erro ao registrar presenças');
    }

    return await response.json();
  } catch (error) {
    console.error('Erro ao registrar presenças:', error);
  }
}

```

```

        return null;
    }
},

async getPresencasByTurmaData(turmaid, data) {
    try {
        const response = await fetch(`${this.API_URL}/presencas/turma/${turmaid}/data/${data}`);
        if (!response.ok) {
            throw new Error('Erro ao buscar presenças');
        }
        return await response.json();
    } catch (error) {
        console.error('Erro ao buscar presenças:', error);
        return [];
    }
},

async getPresencasByAluno(alunoid) {
    try {
        const response = await fetch(`${this.API_URL}/presencas/aluno/${alunoid}`);
        if (!response.ok) {
            throw new Error('Erro ao buscar presenças do aluno');
        }
        return await response.json();
    } catch (error) {
        console.error('Erro ao buscar presenças do aluno:', error);
        return [];
    }
},

formatarData(data) {
    // Helper to format a date as YYYY-MM-DD
    if (data instanceof Date) {
        return data.toISOString().split('T')[0];
    }
    return data;
}
};

```

File: package.json

=====

```
{
  "name": "controle-frequencia",
  "version": "1.0.0",
  "description": "Sistema simples para controle de presença de alunos",
  "main": "server.js",
  "scripts": {
    "start": "node server.js",
    "dev": "nodemon server.js"
  },
  "keywords": [
    "frequencia",
    "alunos",
    "controle"
  ],
  "author": "Marques Vinícius Melo Martins",
  "license": "MIT",
  "dependencies": {
    "cors": "^2.8.5",
    "express": "^4.18.2",
    "uuid": "^9.0.1"
  },
  "devDependencies": {
    "nodemon": "^3.0.1"
  }
}
```

File: css\style.css

=====

