

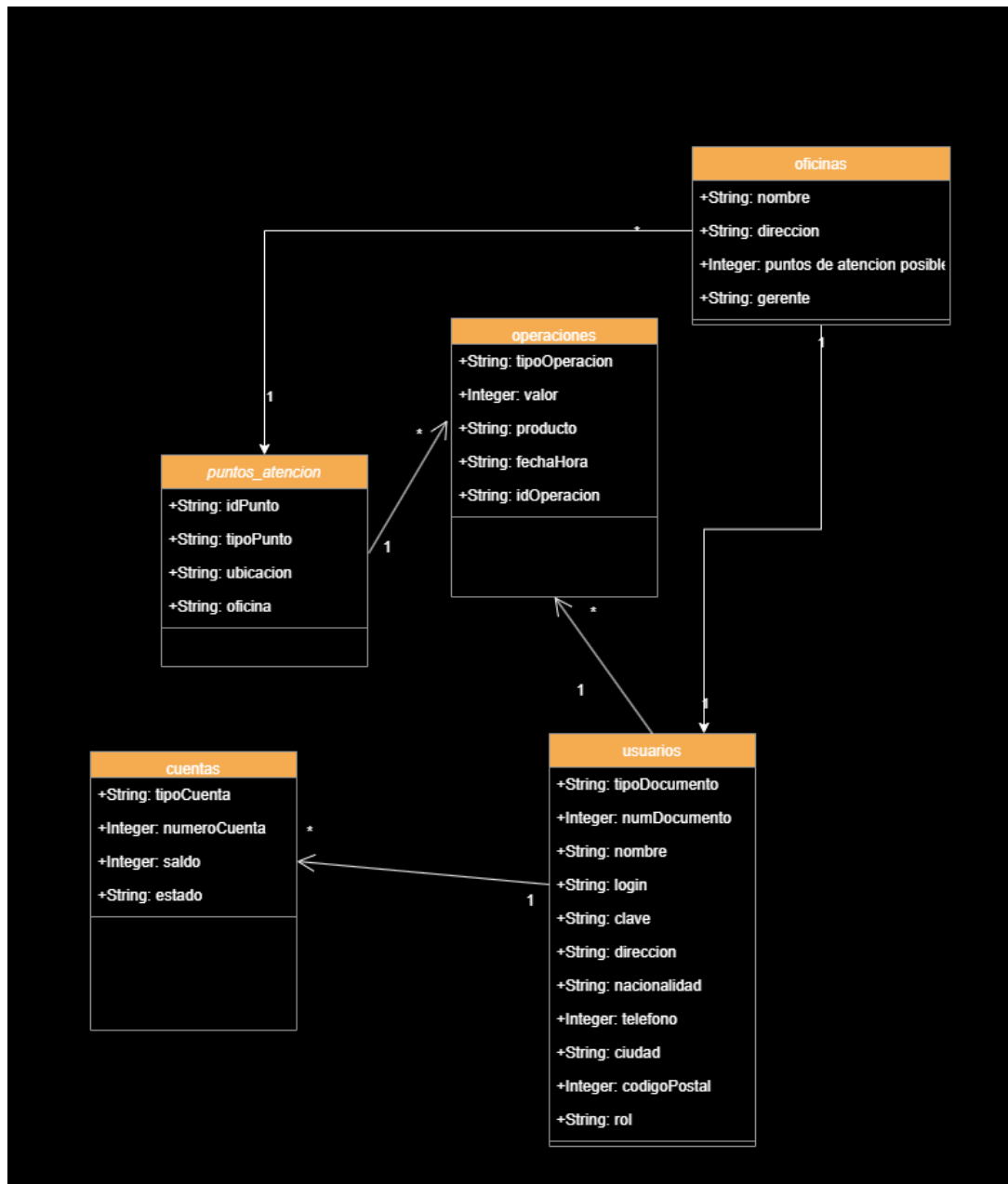
Proyecto – Entrega 4

Grupo 6

Contenidos

- I. Análisis y modelo conceptual
- II. Diseño de la base de datos
 - a. Análisis de la carga de trabajo (workload)
 - b. Descripción de entidades de datos y las relaciones
 - c. Scripts

I. Análisis



II. Diseño de la base de datos

a. Análisis de la carga de trabajo (workload)

Entidades y sus atributos

Las entidades de la base de datos con sus atributos son las siguientes:

1. Usuarios

id: int

nombre: string (obligatorio)
login: string (obligatorio)
palabra_clave: string (obligatorio)
nacionalidad: string (obligatorio)
direccion: string (obligatorio)
telefono: string (obligatorio)
ciudad: string (obligatorio)
codigo_postal: string (obligatorio)
rol: string (obligatorio)

2. Oficinas

Puntos de atencion posibles: int (obligatorio)
nombre: string (obligatorio)
direccion: string (obligatorio)
gerente: string (obligatorio)

3. Puntos de Atención

id: int (obligatorio)
tipo_punto: string (obligatorio)
ubicación: string (obligatorio)
oficina: string (obligatorio)

4. Cuentas

id: int (obligatorio)
tipo: string (obligatorio)
estado: string (obligatorio)
saldo: int (obligatorio)
cliente_id: int (obligatorio)
oficina: int (obligatorio)

5. Operaciones

id: int (obligatorio)
cuenta: int (obligatorio)
tipo: string (obligatorio)
id_usuario: int (obligatorio)
valor: int (obligatorio)
fecha_hora: string (obligatorio)

Cuantificación de las entidades

1. Usuarios: Para los usuarios, según el estimado de operaciones de lectura y escritura hechas a diario, debería haber al menos 200 registros para asemejar el funcionamiento que la aplicación tendría normalmente.
2. Oficinas: Para las oficinas, según el estimado de operaciones de lectura y escritura hechas a diario, debería haber al menos 150 registros ya que estos no tienen operaciones de lectura y escritura tan frecuentes.
3. Puntos de atención: Para los puntos de atención, según el estimado de operaciones de lectura y escritura hechas a diario, debería haber al menos 50 registros ya que estos no tienen operaciones de lectura y escritura tan frecuentes.
4. Cuentas: Para las cuentas, según el estimado de operaciones de lectura y escritura hechas a diario, debería haber al menos 500 registros que serán usados para la creación, modificación y lectura de las cuentas. Esto para asemejar el funcionamiento que la aplicación tendría normalmente.
5. Operaciones: Para las operaciones, según el estimado de operaciones de lectura y escritura hechas a diario, debería haber al menos 5000 registros que serán usados para la creación, modificación y lectura de las operaciones. Esto para asemejar el funcionamiento que la aplicación tendría normalmente.

Análisis de las operaciones de lectura y escritura para cada entidad

Entidad	Operaciones de Lectura	Operaciones de Escritura
Usuarios	Consultar perfil de usuario (nombre, login, nacionalidad, direccion, telefono, ciudad, codigo_postal, rol)	Crear usuario, actualizar información de usuario (nombre, login, palabra_clave, nacionalidad, direccion, telefono, ciudad, codigo_postal, rol)
Oficinas	Consultar servicios ofrecidos por oficina (nombre, direccion, gerente, punto_de_atencion_posibles)	Crear oficina, actualizar servicios ofrecidos (nombre, direccion, gerente, punto_de_atencion_posibles)

Puntos de atención	Consultar información del punto de atención (id, tipo_punto, ubicacion, oficina)	Crear punto de atención, actualizar información del punto (id, tipo_punto, ubicacion, oficina)
Cuentas	Consultar estado de la cuenta (id, tipo, estado, saldo, cliente_id, oficina)	Crear cuenta, actualizar estado y saldo de la cuenta (id, tipo, estado, saldo, cliente_id, oficina)
Operaciones	Consultar detalles de la operación (id, cuenta, tipo, id_usuario, valor, fecha_hora)	Crear operación (id, cuenta, tipo, id_usuario, valor, fecha_hora)

Cuantificación de las operaciones de lectura y escritura para cada entidad

Entidad	Cantidad Estimada de Registros	Operaciones de Lectura Diarias Estimadas	Operaciones de Escritura Diarias Estimadas
Usuarios	200	600 (aproximadamente 3 lecturas por usuario)	200

Oficinas	150	150 (aproximadamente 1 lectura por oficina cada semana)	5 escrituras mensuales
Puntos de atención	50	50 (aproximadamente 1 lectura por punto por semana)	5 escrituras mensuales
Cuentas	500	5000 (aproximadamente 10 lecturas por cuenta)	100
Operaciones	5000	5000 (1 lectura por operación)	5000

b. Descripción de entidades de datos y las relaciones

Lista de Entidades y Descripción

Oficinas

Atributos:

nombre (String): Nombre de la oficina.

direccion (String): Dirección de la oficina.

puntosDeAtencionPosibles (Integer): Número de puntos de atención posibles.

gerente (String): Nombre del gerente de la oficina.

Descripción: Representa una oficina bancaria.

Puntos_Atencion

Atributos:

idPunto (String): Identificador único del punto de atención.

tipoPunto (String): Tipo de punto de atención.

ubicacion (String): Ubicación del punto de atención.

oficina (String): Identificador de la oficina a la que pertenece.

Descripción: Representa un punto de atención dentro de una oficina bancaria.

Operaciones

Atributos:

tipoOperacion (String): Tipo de operación realizada.

valor (Integer): Valor de la operación.

producto (String): Producto relacionado con la operación.

fechaHora (String): Fecha y hora de la operación.

idOperacion (String): Identificador único de la operación.

idPunto (String): Identificador del punto de atención donde se realizó la operación.

Descripción: Representa una operación bancaria.

Cuentas

Atributos:

tipoCuenta (String): Tipo de cuenta (e.g., ahorro, corriente).

numeroCuenta (Integer): Número de cuenta.

saldo (Integer): Saldo de la cuenta.

estado (String): Estado de la cuenta.

numDocumento (Integer): Número de documento del usuario propietario de la cuenta.

Descripción: Representa una cuenta bancaria.

Usuarios

Atributos:

tipo Documento (String): Tipo de documento del usuario (e.g., DNI, pasaporte).

numDocumento (Integer): Número del documento.

nombre (String): Nombre del usuario.

login (String): Nombre de usuario para login.

clave (String): Contraseña del usuario.

direccion (String): Dirección del usuario.

nacionalidad (String): Nacionalidad del usuario.

telefono (Integer): Teléfono del usuario.

ciudad (String): Ciudad del usuario.

codigoPostal (Integer): Código postal del usuario.

rol (String): Rol del usuario (e.g., administrador, cliente).

Descripción: Representa a un usuario del sistema bancario.

Relaciones entre Entidades y su Cardinalidad

Oficinas – Puntos_Atencion

Cardinalidad: Uno a muchos (1)

Tipo de Asociación: Referenciada (PuntoAtencion contiene una referencia a Oficina)

Puntos_Atencion - Operaciones

Cardinalidad: Uno a muchos (1)

Tipo de Asociación: Referenciada (Operacion contiene una referencia a PuntoAtencion)

Usuarios - Cuentas

Cardinalidad: Uno a muchos (1)

Tipo de Asociación: Referenciada (Cuenta contiene una referencia a Usuario)

Usuario - Oficinas

Cardinalidad: Uno a muchos (1)

Tipo de Asociación: Referenciada (Oficina contiene una referencia a Usuario)

Análisis de Selección de Esquema de Asociación

Relación	Cardinalidad	Asociación	Justificación
Oficinas – Puntos_atencion	1	Referenciada	La relación referenciada permite mantener la oficina separada de los puntos de atención, lo cual es útil si los puntos de atención cambian con frecuencia.
Puntos_Atencion - Operaciones	1	Referenciada	Similarmente, las operaciones son numerosas y frecuentemente cambiantes, lo que hace más eficiente tener referencias en lugar de documentos embebidos.
Usuarios - Cuentas	1	Referenciada	Referenciar las cuentas desde el usuario permite manejar cambios en las cuentas sin necesidad de modificar los documentos del usuario.
Usuarios - Oficinas	1	Referenciada	Referenciar los usuarios desde la oficina permite manejar los empleados que están relacionados a la oficina

Descripción Gráfica usando JSON

1. Oficinas – Puntos_Atencion (Referenciada)

```
{  
  "oficinas": {  
    "nombre": "Oficina Central",  
    "direccion": "Av. Principal 123",  
    "puntosDeAtencionPosibles": 5,  
    "gerente": "Juan Pérez"  
  },  
  "puntos_atencion": {
```



```
"idPunto": "PA001",
"tipoPunto": "Cajero",
"ubicacion": "Hall principal",
"oficina": "Oficina Central"
}
}

2. Puntos_Atencion - Operaciones (Referenciada)
```

```
{
  "puntos_atencion": {
    "idPunto": "PA001",
    "tipoPunto": "Cajero",
    "ubicacion": "Hall principal",
    "oficina": "Oficina Central"
  },
  "operaciones": {
    "idOperacion": "OP001",
    "tipoOperacion": "Depósito",
    "valor": 1000,
    "producto": "Cuenta de Ahorros",
    "fechaHora": "2024-05-25T10:30:00Z",
    "idPunto": "PA001"
  }
}
```

3. Usuarios - Cuentas (Referenciada)

```
{
  "usuarios": {
    "tipoDocumento": "DNI",
    "numDocumento": 12345678,
    "nombre": "María González",
    "login": "maria_g",
    "clave": "password123",
    "direccion": "Calle Falsa 456",
    "nacionalidad": "Argentina",
    "telefono": 987654321,
    "ciudad": "Buenos Aires",
  }
}
```

```
"codigoPostal": 1234,
"rol": "cliente"
},
"cuentas": {
  "tipoCuenta": "Ahorro",
  "numeroCuenta": 987654321,
  "saldo": 5000,
  "estado": "activa",
  "numDocumento": 12345678
}
}
```

3. Usuarios - Oficinas (Referenciada)

```
{
  "usuarios": {
    "tipoDocumento": "DNI",
    "numDocumento": 12345678,
    "nombre": "Juan Pérez",
    "login": "juan_p",
    "clave": "password123",
    "direccion": "Calle Falsa 456",
    "nacionalidad": "Argentina",
    "telefono": 987654321,
    "ciudad": "Buenos Aires",
    "codigoPostal": 1234,
    "rol": "gerente"
  },
  {
    "oficinas": {
      "nombre": "Oficina Central",
      "direccion": "Av. Principal 123",
      "puntosDeAtencionPosibles": 5,
      "gerente": "Juan Pérez"
    }
  }
}
```

c. Scripts

Creación de usuarios

```
db.createCollection("usuarios", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["nombre", "login", "palabra_clave", "nacionalidad", "direccion", "telefono", "ciudad",
"codigo_postal", "rol"],
      properties: {
        id: {
          bsonType: "int"
        },
        nombre: {
          bsonType: "string"
        },
        login: {
          bsonType: "string"
        },
        palabra_clave: {
          bsonType: "string"
        },
        nacionalidad: {
          bsonType: "string"
        },
        direccion: {
          bsonType: "string"
        },
        telefono: {
          bsonType: "string"
        },
        ciudad: {
          bsonType: "string"
        },
        codigo_postal: {
          bsonType: "string"
        },
        rol: {
          bsonType: "string"
        }
      }
    }
  }
});
```

Creación de oficinas

```
db.createCollection("oficinas", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["nombre", "direccion", "puntosDeAtencionPosibles", "gerente"],
      properties: {
        nombre: {
```

```

        bsonType: "string"
    },
    direccion: {
        bsonType: "string"
    },
    puntosDeAtencionPosibles: {
        bsonType: "int"
    },
    gerente: {
        bsonType: "string"
    }
}
}
}
});

```

Creación de puntos de atención

```

db.createCollection("puntos_atencion", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: [ "tipoPunto", "ubicacion", "oficina" ],
      properties: {
        tipoPunto: {
          bsonType: "string"
        },
        ubicacion: {
          bsonType: "string"
        },
        oficina: {
          bsonType: "string"
        }
      }
    }
  }
});

```

```
});
```

Creación de cuentas

```
db.createCollection("cuentas", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["tipoCuenta", "numeroCuenta", "saldo", "estado", "numDocumento"],
      properties: {
        tipoCuenta: {
          bsonType: "string"
        },
        numeroCuenta: {
          bsonType: "int"
        },
        saldo: {
          bsonType: "int"
        },
        estado: {
          bsonType: "string"
        },
        numDocumento: {
          bsonType: "int"
        }
      }
    }
  }
});
```

Creación de operaciones

```
db.createCollection("operaciones", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["id", "cuenta", "tipo", "id_usuario", "valor", "fecha_hora"],
      properties: {
        id: {
          bsonType: "int"
        },
        cuenta: {
          bsonType: "int"
        }
      }
    }
  }
});
```

```
    },
    tipo: {
      bsonType: "string"
    },
    id_usuario: {
      bsonType: "int"
    },
    valor: {
      bsonType: "int"
    },
    fecha_hora: {
      bsonType: "string"
    }
  }
}
});
```