

Entrega 3 Sistemas Transaccionales

3) Diseño Base de Datos Actualizar este punto con el UML actualizado y sus respectivos atributos

A) Análisis de la carga de trabajo (workload)

a) Identificar entidades y atributos

Cliente:

- id: Integer
- tipo: String
- nombre: String
- num_doc: String
- tipo_doc: String
- direccion: String
- mail: String
- telefono: String
- ciudad: String
- estado: String
- clave: String
- login: String
- nacionalidad: String
- dept: String
- codigo_post: String

Cuenta:

- id: Integer
- tipo: String
- saldo: Integer
- estado: String
- cliente: Cliente
- ultima_trans: Date (Fecha última transacción)

OperacionBancariaCuenta:

- id: Integer
- cuenta: Cuenta
- tipo: String
- fecha: Date
- valor: Integer

Transferencia:

- id: Integer
- id_cuenta_origen: Integer
- id_cuenta_destino: Integer
- valor: Integer
- fecha: Date

Empleado:

- id: Integer
- cargo: String

- nombre: String
- num_doc: String
- tipo_doc: String
- direccion: String
- mail: String
- telefono: String
- ciudad: String
- estado: String
- clave: String
- login: String
- nacionalidad: String
- departamento: String
- codigo_postal: String (definido en el UML)
- tipo: String
- idOficina: String

Oficina:

- id: Integer
- nombre: String
- direccion: String
- id_gerente: Integer
- num_pa: Integer (número de puntos de atención)

PuntoAtencion:

- id: Integer
- tipo: Integer
- oficina: Oficina

b)

Entidades	Operación	Información necesaria	Tipo	Tasa
Oficinas	Crear/Modificar Oficina	Detalles de la oficina	Write	1/mes
Oficinas	Consultar Oficina	Detalles de la oficina	Read	1/semana
Puntos de Atención	Crear/Modificar Punto de Atención	Tipo, Localización	Write	1/mes
Puntos de Atención	Consultar Punto de Atención	Tipo, Localización	Read	1/semana
Clientes	Crear/Modificar	Detalles del	Write	200/dia

	Cliente	cliente		
Cientes	Consultar Cliente	Detalles del cliente	Read	500/dia
Empleados	Crear/Modificar Empleado	Detalles del usuario	Write	200/dia
Empleados	Consultar Empleado	Detalles del usuario	Read	500/dia
Cuentas	Crear/Modificar Cuenta	Tipo, saldo, estado	Write	500/dia
Cuentas	Consultar Cuenta	Tipo, saldo, estado	Read	5000/dia
Operaciones Bancarias	Registrar Operación sobre cuenta	Detalles de la operación	Write	20000/dia
Operaciones Bancarias	Consultar Operación sobre cuenta	Detalles de la operación	Read	5000/dia

c) Analicen las operaciones de lectura y escritura para cada entidad.

Entidades	Operación	Información necesaria	Tipo
Oficinas	Crear/Modificar Oficina	Detalles de la oficina	Write
Oficinas	Consultar Oficina	Detalles de la oficina	Read
Puntos de Atención	Crear/Modificar Punto de Atención	Tipo, Localización	Write
Puntos de Atención	Consultar Punto de Atención	Tipo, Localización	Read
Cientes	Crear/Modificar Cliente	Detalles del cliente	Write
Cientes	Consultar Cliente	Detalles del cliente	Read
Empleados	Crear/Modificar Empleado	Detalles del usuario	Write

Empleados	Consultar Empleado	Detalles del usuario	Read
Cuentas	Crear/Modificar Cuenta	Tipo, saldo, estado	Write
Cuentas	Consultar Cuenta	Tipo, saldo, estado	Read
Operaciones Bancarias	Registrar Operación sobre cuenta	Detalles de la operación	Write
Operaciones Bancarias	Consultar Operación sobre cuenta	Detalles de la operación	Read

d)

Entidades	Operación	Información necesaria	Tipo	Tasa
Oficinas	Crear/Modificar Oficina	Detalles de la oficina	Write	1/mes
Oficinas	Consultar Oficina	Detalles de la oficina	Read	1/semana
Puntos de Atención	Crear/Modificar Punto de Atención	Tipo, Localización	Write	1/mes
Puntos de Atención	Consultar Punto de Atención	Tipo, Localización	Read	1/semana
Clientes	Crear/Modificar Cliente	Detalles del cliente	Write	200/dia
Clientes	Consultar Cliente	Detalles del cliente	Read	500/dia
Empleados	Crear/Modificar Empleado	Detalles del usuario	Write	200/dia
Empleados	Consultar Empleado	Detalles del usuario	Read	500/dia
Cuentas	Crear/Modificar Cuenta	Tipo, saldo, estado	Write	500/dia
Cuentas	Consultar Cuenta	Tipo, saldo, estado	Read	5000/dia
Operaciones Bancarias	Registrar Operación	Detalles de la operación	Write	20000/dia

	sobre cuenta			
Operaciones Bancarias	Consultar Operación sobre cuenta	Detalles de la operación	Read	5000/dia

B)

a. Lista de Entidades con Descripción Actualizar Atributos con los actualizados

Cliente:

Descripción: Representa a los usuarios del banco, ya sean personas naturales o jurídicas. Un cliente puede ser también un empleado del banco.

Atributos:

- id: Integer
- tipo: String (natural o jurídico)
- nombre: String
- num_doc: String (número de documento)
- tipo_doc: String (tipo de documento)
- direccion: String
- mail: String (correo electrónico)
- telefono: String
- ciudad: String
- estado: String (activo/inactivo)
- clave: String
- login: String
- nacionalidad: String
- dept: String
- codigo_post: String

Cuenta:

Descripción: Representa las cuentas bancarias abiertas por los clientes. Pueden ser de varios tipos como ahorros, corriente y AFC (ahorro fomento de construcción).

Atributos:

- id: Integer
- numero: Integer (número de cuenta)
- tipo: String
- saldo: Integer
- estado: String (activa, cerrada, desactivada)
- cliente: Cliente
- ultima_trans: Date (Fecha última transacción)

OperacionBancariaCuenta:

Descripción: Representa las transacciones financieras realizadas sobre una cuenta específica, como depósitos o retiros.

Atributos:

- id: Integer
- cuenta: Cuenta

- tipo: String (tipo de operación)
- fecha: Date
- valor: Integer

Transferencia:

Descripción: Subtipo de operación bancaria que implica mover fondos de una cuenta a otra dentro del mismo banco.

Atributos:

- id: Integer
- id_cuenta_origen: Integer (cuenta origen)
- id_cuenta_destino: Integer (cuenta destino)
- valor: Integer
- fecha: Date

Empleado:

Descripción: Representa a los trabajadores del banco, incluyendo roles como gerentes de oficina, cajeros, y el gerente general.

Atributos:

- id: Integer
- cargo: String
- nombre: String
- num_doc: String (número de documento)
- tipo_doc: String (tipo de documento)
- direccion: String
- mail: String (correo electrónico)
- telefono: String
- ciudad: String
- estado: String (activo/inactivo)
- clave: String
- login: String
- nacionalidad: String
- departamento: String
- codigo_postal: String

Oficina:

Descripción: Representa una ubicación física del banco donde se atienden clientes.

Atributos:

- id: Integer
- nombre: String
- direccion: String
- id_gerente: Integer (referencia a Empleado)
- num_pa: Integer (número de puntos de atención)

PuntoAtencion:

Descripción: Representa los diferentes puntos donde los clientes pueden realizar operaciones bancarias, como cajeros automáticos o puntos de atención personalizada en una oficina.

Atributos:

- id: Integer
- tipo: Integer (personalizado, automático, digital)
- oficina: Oficina (referencia a Oficina)

b. Relaciones entre Entidades y su Cardinalidad

Cliente - Cuenta

- Relación: Un cliente puede tener múltiples cuentas.
- Cardinalidad: Uno a muchos.

Cuenta - Operacion Bancaria Cuenta

- Relación: Una cuenta puede tener múltiples operaciones bancarias.
- Cardinalidad: Uno a muchos.

Cuenta - Transferencia

- Relación: Una cuenta puede ser origen o destino en múltiples transferencias.
- Cardinalidad: Muchos a muchos (mediante dos relaciones uno a muchos, una para origen y otra para destino).

Empleado - Oficina **revisar esto**

- Relación: Una oficina puede tener muchos empleados, pero un empleado solo una oficina
- Cardinalidad: Uno a muchos.

Oficina - PuntoAtencion

- Relación: Una oficina puede tener múltiples puntos de atención.
- Cardinalidad: Uno o cero a muchos.

c.

Relaciones	Tipo	Justificación
Cliente - Cuenta	Referencial	<p>Simplicity: No, mantener Cliente y cuenta juntos no simplifica el data model.</p> <p>Go Together: Sí, un cliente si tiene cuentas.</p> <p>Query Atomicity: No, en los requerimientos Cuenta y Cliente se consultan por separado.</p> <p>Update Complexity: No, cuentas se actualiza mucho más que clientes y se hace por separado.</p> <p>Archival: No, primero se crea un cliente y luego una cuenta.</p> <p>Cardinality: Sí, cada cliente puede tener muchas cuantas.</p> <p>Data Duplication: No, si cada cliente tuviera sus</p>

		<p>cuentas embebidas, no habría duplicación de información porque una cuenta solo puede tener un dueño.</p> <p>Document Size: No, sería la misma.</p> <p>Document Growth: Sí, un cliente puede tener infinitas cuentas.</p> <p>Workload: Sí, las cuentas y los clientes se crean por separado.</p> <p>Individuality: No, una cuenta no puede existir sin un cliente.</p>
Cuenta - OBC	Referencial	<p>Simplicity: No, mantener Cuenta y OBC juntos no simplifica el data model.</p> <p>Go Together: Sí, una cuenta si tiene OBCs.</p> <p>Query Atomicity: No, en los requerimientos Cuenta y OBC se consultan por separado.</p> <p>Update Complexity: No, OBC se actualiza mucho más que cuentas y se hace por separado.</p> <p>Archival: No, primero se crea una cuenta y luego un OBC.</p> <p>Cardinality: Sí, cada cuenta puede tener muchos OBCs.</p> <p>Data Duplication: No, si cada cuenta tuviera sus OBC embebidos, no habría duplicación de información porque un OBC solo puede tener una cuenta.</p> <p>Document Size: No, sería la misma.</p> <p>Document Growth: Sí, una cuenta puede tener infinitos OBCs.</p> <p>Workload: Sí, las cuentas y los OBCs se crean por separado.</p> <p>Individuality: No, un OBC no puede existir sin una cuenta.</p>
Cuenta - Transferencia	Referencial	<p>Simplicity: No, mantener Cuenta y Transferencia juntos no simplifica el data model.</p> <p>Go Together: Sí, una cuenta si tiene transferencias.</p> <p>Query Atomicity: No, en los requerimientos Cuenta y Transferencias se consultan por separado.</p> <p>Update Complexity: No, transferencia se actualiza mucho más que cuenta y se hace por separado.</p> <p>Archival: No, primero se crea una cuenta y luego una transferencia.</p> <p>Cardinality: Sí, cada cuenta puede tener muchas transferencias.</p>

		<p>Data Duplication: Sí, si cada cuenta tuviera sus transferencias embebidas, habría duplicación porque cada transferencia estaría dos veces, una por cada cuenta.</p> <p>Document Size: Si, aumentaria.</p> <p>Document Growth: Sí, una cuenta puede tener infinitas transferencias.</p> <p>Workload: Sí, las cuentas y los OBCs se crean por separado.</p> <p>Individuality: No, una transferencia no puede existir sin una cuenta.</p>
Oficina - PA	Referencial	<p>Simplicity: No, mantener Oficina y PA juntos no simplifica el data model.</p> <p>Go Together: Sí, una Oficina si tiene PAs.</p> <p>Query Atomicity: No, en los requerimientos Oficina y PAs se consultan por separado.</p> <p>Update Complexity: No, PAs se actualiza mucho más que oficina y se hace por separado.</p> <p>Archival: No, primero se crea una oficina y luego un PA.</p> <p>Cardinality: Sí, cada Oficina puede tener muchas PAs.</p> <p>Data Duplication: No, si cada oficina tuviera sus PAs embebidos, no habría duplicación de información porque un PA solo puede tener una oficina.</p> <p>Document Size: No, sería la misma.</p> <p>Document Growth: Sí, una oficina puede tener infinitos PAs.</p> <p>Workload: Sí, las oficinas y los PAs se crean por separado.</p> <p>Individuality: Si, un PA puede existir sin una oficina solo si es virtual.</p>
Oficina - Empleado	Referencial	<p>Simplicity: No, mantener Oficina y Empleado juntos no simplifica el data model.</p> <p>Go Together: Sí, una Oficina si tiene Empleados.</p> <p>Query Atomicity: No, en los requerimientos Oficina y Empleados se consultan por separado.</p> <p>Update Complexity: No, Empleados se actualiza mucho más que oficina y se hace por separado.</p> <p>Archival: No, primero se crea una oficina y luego un Empleado.</p> <p>Cardinality: Sí, cada Oficina puede tener muchos empleados.</p> <p>Data Duplication: No, si cada oficina tuviera</p>

		<p>empleados sus embebidos, no habría duplicación de información porque un empleado solo puede tener una oficina.</p> <p>Document Size: No, sería la misma.</p> <p>Document Growth: Sí, una oficina puede tener infinitos empleados.</p> <p>Workload: Sí, las oficinas y los empleados se crean por separado.</p> <p>Individuality: No, un empleado no puede existir sin una oficina en este caso. (pero en la vida real sí).</p>
--	--	---

d. Está en un PDF en el Proyecto.

C) El punto 3c esta en una carpeta a parte dentro del proyecto.

ESCENARIOS DE PRUEBA:

Parte 1

https://drive.google.com/file/d/1bllt6zk_Gz1LqAbZcol2PO_Grka8SpJY/view?usp=sharing

Parte 2:

<https://drive.google.com/file/d/1mHQAqKGHOsR7JgEtw2tGtSWbvQrReNhN/view?usp=sharing>