

Proyecto 1

@RONY ORMANDY ORTÍZ ALVAREZ

Proyecto 1 Individual: Sistema Bancario - Banco JP Morgan

Introducción

Objetivo General

Objetivos Específicos

Definición del Sistema

Estructura Organizacional del Banco JP Morgan

Estructura Operacional

Operación General del Banco JP Morgan

Requisitos Funcionales

Requisitos No Funcionales

Diagrama de Casos de Uso

Descripción de los Casos de Uso:

Explicación del Diagrama:

Estimación de Capacidad

Diseño de Bajo Nivel

Explicación del Diagrama:

Diseño de Alto Nivel

Descripción de los Componentes Principales:

Explicación del Diagrama:

Características de la Arquitectura Monolítica:

Consultas a la Base de Datos

1. Consultas Relacionadas con Clientes

2. Consultas Relacionadas con Transacciones

3. Consultas Relacionadas con Préstamos

4. Consultas Relacionadas con Tarjetas de Crédito

5. Consultas Relacionadas con Empleados

6. Consultas Relacionadas con Sucursales y Agencias

7. Consultas de Auditoría y Reportes

8. Consultas de Seguridad

Implementación de Código API

Datos Disponibles

Proyecto 1 Individual: Sistema Bancario - Banco JP Morgan

Introducción

Este proyecto tiene como objetivo permitir que los estudiantes de Ingeniería en Ciencias y Sistemas trabajen en la creación y gestión de un sistema bancario basado en bases de datos SQL. El sistema estará basado en un banco ficticio denominado **Banco JP Morgan**, que replica las operaciones de grandes bancos en Guatemala. A lo largo del proyecto, los estudiantes abordarán varios aspectos fundamentales de bases de datos, incluyendo diseño, normalización, consultas avanzadas, y manejo de transacciones, todo dentro de un contexto realista y práctico.

Objetivo General

El objetivo principal del proyecto es que los estudiantes se familiaricen con el diseño y manejo de bases de datos SQL en un contexto bancario, desarrollando habilidades clave como la normalización, la gestión de transacciones, la creación de consultas avanzadas y la optimización para manejar grandes volúmenes de datos en un entorno de producción.

Objetivos Específicos

1. **Diseño de Base de Datos:** El estudiante debe diseñar una base de datos completamente normalizada que represente las operaciones bancarias de un banco real.
2. **Gestión de Transacciones:** Desarrollar la capacidad de manejar transacciones bancarias de manera eficiente y segura, garantizando la integridad de los datos.
3. **Consultas SQL Avanzadas:** Permitir a los estudiantes escribir consultas complejas que involucren múltiples tablas, manejo de fechas, agregaciones y optimización de rendimiento.
4. **Soporte de Cargas Simultáneas:** Diseñar y validar el sistema para que soporte múltiples operaciones simultáneas, asegurando la integridad de los datos y un rendimiento adecuado.

5. **Seguridad de Datos:** Establecer mecanismos de seguridad robustos, incluyendo controles de acceso a la base de datos y manejo seguro de las transacciones.
6. **Modelado de Base de Datos:** Aplicar las mejores prácticas de modelado de datos en la construcción de bases de datos transaccionales, asegurando la integridad referencial y la optimización de consultas.

Definición del Sistema

Banco JP Morgan es un banco ficticio que ofrece una amplia gama de servicios financieros, incluyendo cuentas de ahorro, préstamos, tarjetas de crédito y débito, además de servicios de pago y transferencias. El banco tiene presencia en todo el país y cuenta con una amplia red de sucursales y agencias que permiten ofrecer servicios tanto en áreas urbanas como rurales.

Estructura Organizacional del Banco JP Morgan

El Banco JP Morgan se organiza mediante un organigrama que refleja tanto la jerarquía ejecutiva como las funciones operativas y de apoyo dentro del banco. A continuación, se describe cómo está estructurado y cómo opera:

1. Nivel Ejecutivo:

- **Presidente del Consejo de Administración:** Responsable de la estrategia general y la supervisión del cumplimiento de las políticas establecidas. Colabora con el Gerente General para asegurar que la visión y misión del banco se cumplan.
- **Gerente General:** Supervisa las operaciones diarias del banco y la implementación de las estrategias aprobadas por el Consejo de Administración, asegurando que todos los departamentos trabajen en conjunto para alcanzar los objetivos del banco.

2. Dirección de Banca:

- **Dirección de Banca de Personas:** Administra los productos y servicios para clientes individuales, como cuentas de ahorro, préstamos personales y tarjetas de crédito.

- **Dirección de Banca Empresarial:** Responsable de los servicios financieros para empresas, como créditos empresariales, servicios de comercio exterior, y productos financieros diseñados para apoyar el crecimiento empresarial.

3. Dirección Operativa y de Riesgos:

- **Dirección de Operaciones:** Administra la red de sucursales y cajeros automáticos, supervisando las operaciones bancarias en todo el país.
- **Dirección de Riesgos:** Gestiona los riesgos financieros, operacionales y de crédito, protegiendo al banco de posibles pérdidas y asegurando la estabilidad financiera.

4. Dirección Financiera y Tecnológica:

- **Dirección Financiera:** Planifica las finanzas, controla el presupuesto y gestiona la tesorería del banco.
- **Dirección de Tecnología y Sistemas:** Supervisa la infraestructura tecnológica, los sistemas de información y la seguridad informática del banco.

5. Dirección de Recursos Humanos y Marketing:

- **Dirección de Recursos Humanos:** Administra el talento humano, incluyendo la contratación, capacitación y desarrollo de empleados.
- **Dirección de Marketing y Comunicaciones:** Promueve los productos y servicios del banco, gestiona la imagen corporativa y mantiene relaciones públicas.

6. Operaciones en Sucursal:

- **Gerentes de Sucursal:** Supervisan las operaciones diarias, aseguran el cumplimiento de los objetivos de ventas y brindan un excelente servicio al cliente.
- **Asesores Financieros:** Ofrecen asesoría sobre productos financieros como préstamos, inversiones y seguros.
- **Ejecutivos de Cuenta:** Gestionan las cuentas de clientes clave, proporcionando un servicio personalizado.

- **Cajeros y Personal de Ventanilla:** Realizan transacciones diarias, incluyendo depósitos, retiros y pagos de servicios.

Estructura Operacional

- **Clientes:** 1 millón de clientes registrados en la base de datos, con cuentas monetarias, cuentas de ahorro y cuentas corrientes.
- **Transacciones:** El banco procesa 500,000 transacciones, que incluyen depósitos, retiros, transferencias y pagos.
- **Préstamos:** 100,000 préstamos otorgados con montos que varían entre 5,000 y 1,000,000 de moneda nacional, y tasas de interés entre 3% y 15%.
- **Tarjetas de Crédito:** 100,000 tarjetas emitidas, cada una con un límite de crédito, saldo, fecha de emisión, fecha de expiración, y ciclos de facturación.
- **Sucursales/Agencias:** 1,100 sucursales y 3,000 agencias distribuidas en todo el país, cada una con su propio horario de atención y servicios.
- **Empleados:** 11,000 empleados desempeñan roles que van desde gerentes de sucursal hasta cajeros y asesores financieros.

Operación General del Banco JP Morgan

El Banco JP Morgan opera a través de una extensa red de sucursales y cajeros automáticos distribuidos en todo el país. Facilita transacciones como depósitos, retiros, pagos de servicios, y transferencias, además de ofrecer productos financieros como préstamos y seguros. La tecnología y los sistemas de información son fundamentales para la operación del banco, permitiendo una gestión eficiente de los recursos y garantizando la seguridad de las transacciones.

Requisitos Funcionales

1. **Registro de usuarios:** Permitir a los usuarios crear sus cuentas de forma segura, recopilando información necesaria para la identificación y comunicación.
2. **Procesamiento de pagos:** Soportar varios métodos de pago (tarjetas de crédito, débito, transferencias bancarias) garantizando un procesamiento

eficiente de las transacciones.

3. **Historial de transacciones:** Mantener un registro completo de las transacciones de los usuarios, permitiendo que accedan y revisen su historial.
4. **Autenticación y autorización de usuarios:** Implementar un sistema robusto para la autenticación y autorización en diversas operaciones del sistema.
5. **Gestión de Cuentas:** Crear, actualizar y consultar cuentas de clientes. Realizar operaciones de depósitos, retiros y transferencias entre cuentas.
6. **Procesamiento de Préstamos:** Administrar solicitudes de préstamos, procesar pagos y mantener un registro actualizado de los balances.
7. **Gestión de Tarjetas de Crédito:** Registrar y administrar tarjetas de crédito, incluyendo la emisión, pagos de saldo y generación de estados de cuenta.
8. **Consultas SQL:** Ejecutar consultas SQL avanzadas sobre los datos de clientes, transacciones, préstamos y tarjetas de crédito para generar informes financieros y otros análisis.
9. **Seguridad de Datos:** Implementar medidas de seguridad para proteger la información de los clientes y transacciones.

Requisitos No Funcionales

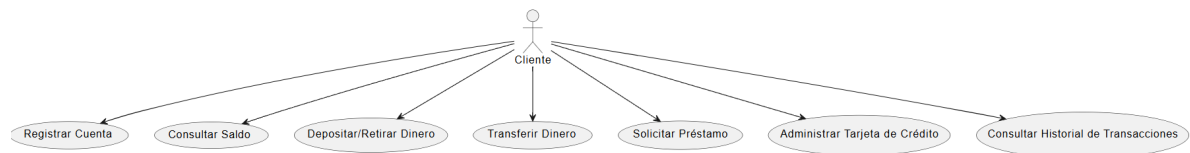
1. **Escalabilidad:** Asegurar que el sistema pueda manejar el crecimiento en el número de clientes, transacciones, préstamos y otros datos.
2. **Disponibilidad:** El sistema debe estar disponible 24/7, con un tiempo de inactividad mínimo.
3. **Rendimiento:** El sistema debe ser capaz de procesar consultas complejas de manera eficiente, incluso con grandes volúmenes de datos.
4. **Seguridad:** Implementar encriptación de datos sensibles y mecanismos de autenticación robustos para el acceso a la base de datos.
5. **Mantenibilidad:** Asegurar que el código del sistema esté organizado y documentado adecuadamente para facilitar futuras modificaciones y ampliaciones.

Diagrama de Casos de Uso

El diagrama de casos de uso proporciona una representación visual de las interacciones que los actores específicos (Cliente, Gerente de Sucursal, Asesor Financiero, Cajero y Administrador) tienen con el sistema bancario del **Banco JP Morgan**. Cada rol está asociado a diferentes responsabilidades dentro del banco.

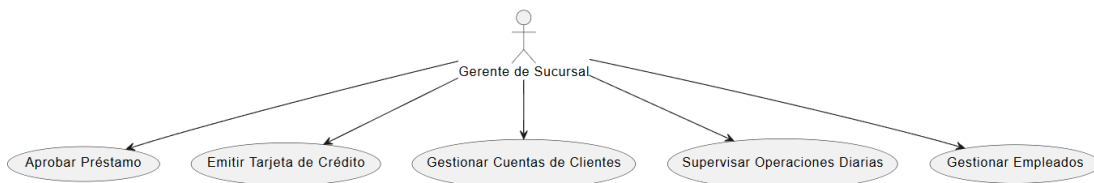
Descripción de los Casos de Uso:

1. Cliente:



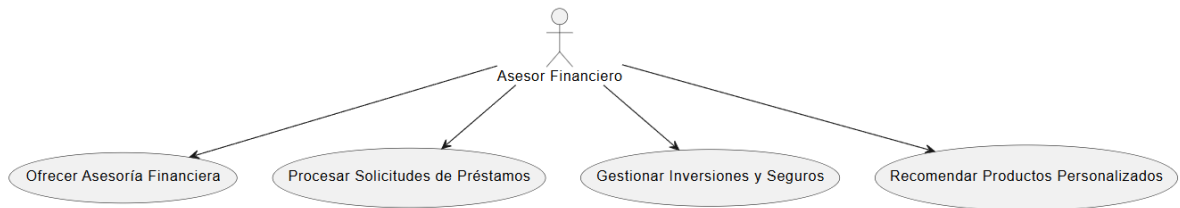
- Registrar una nueva cuenta.
- Consultar saldo de cuentas.
- Realizar depósitos y retiros.
- Transferir dinero entre cuentas.
- Solicitar préstamos.
- Administrar tarjetas de crédito.
- Consultar historial de transacciones.

2. Gerente de Sucursal:



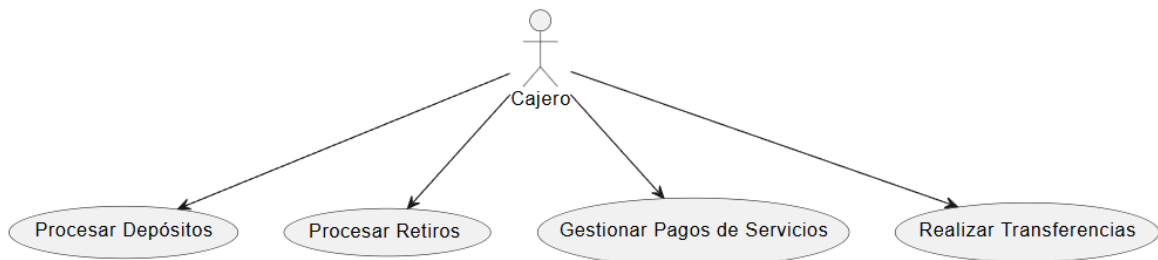
- Aprobar solicitudes de préstamos.
- Emitir tarjetas de crédito.
- Gestionar cuentas de clientes.
- Supervisar las operaciones diarias de la sucursal.
- Asignar y gestionar empleados.

3. Asesor Financiero:



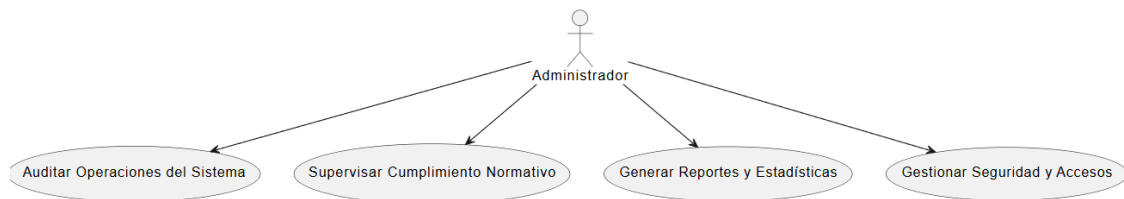
- Ofrecer asesoría sobre productos financieros.
- Procesar solicitudes de préstamos.
- Gestionar inversiones y seguros para los clientes.
- Proporcionar recomendaciones personalizadas basadas en el perfil del cliente.

4. Cajero:



- Procesar depósitos y retiros de clientes.
- Gestionar pagos de servicios.
- Realizar transferencias entre cuentas.

5. Administrador:



- Auditar todas las operaciones del sistema.

- Supervisar el cumplimiento normativo.
- Gestionar los reportes de rendimiento y las estadísticas del banco.
- Gestionar la seguridad del sistema y los accesos.

Explicación del Diagrama:

1. **Cliente:** El cliente interactúa directamente con el sistema para registrar cuentas, consultar saldos, realizar transacciones, solicitar préstamos y administrar tarjetas de crédito. Es el usuario final del sistema.
2. **Gerente de Sucursal:** El gerente de sucursal tiene un rol clave en la supervisión y aprobación de operaciones bancarias. Es responsable de aprobar solicitudes de préstamos, emitir tarjetas de crédito, gestionar cuentas de clientes, y también asignar empleados y supervisar las operaciones diarias en la sucursal.
3. **Asesor Financiero:** El asesor financiero se enfoca en la relación con los clientes, ayudándolos con la gestión de inversiones, seguros y préstamos. Proporciona asesoría sobre productos financieros y realiza recomendaciones personalizadas.
4. **Cajero:** El cajero es responsable de las operaciones cotidianas de caja, como procesar depósitos y retiros, gestionar pagos de servicios, y realizar transferencias entre cuentas.
5. **Administrador:** El administrador tiene un rol de supervisión sobre todo el sistema, gestionando auditorías, asegurando el cumplimiento normativo, generando reportes, y gestionando la seguridad y los accesos.

Estimación de Capacidad

- **Clientes:** 1 millón de clientes.
- **Tráfico mensual estimado:** Se estima que el sistema manejará aproximadamente 100 millones de operaciones por mes, incluyendo transacciones, pagos, consultas, y registros.

Suponiendo que el 70% del tráfico corresponde a registros de usuarios y pagos:

- **Registros y Pagos:** El 70% de 100 millones de operaciones es igual a 70 millones de operaciones mensuales.
- **Distribución de tráfico:**
 - **Registros de nuevos usuarios:** Se estima que el 30% del tráfico corresponde a registros de nuevos usuarios.
 - **Procesos de pago:** Se estima que el 40% del tráfico corresponde a pago
- **Transacciones por segundo (TPS):** Se estima que el sistema debe soportar un promedio de 10 a 15 transacciones por segundo durante picos de alta demanda. Esto implica:
 - **500,000 transacciones** realizadas a lo largo del año.
 - **TPS promedio diario:** ~6 transacciones por segundo, pero se requiere que el sistema maneje picos más altos, de entre 10 y 15 TPS en horas pico.
- **Almacenamiento por segundo:**
 - Suposición: Cada transacción ocupa 100 KB.
 - Con 70 TPS (combinando registros de usuarios y pagos):
 - **70 transacciones * 100 KB = 7000 KB por segundo** (equivalente a 7 MB/s).
- **Almacenamiento requerido por año:**
 - **7 MB por segundo * 60 segundos * 60 minutos * 24 horas * 365 días = 221 TB por año.**
- **IOPS (Input/Output Operations Per Second):** Estimado para la base de datos basada en la cantidad de operaciones de lectura/escritura:
 - **Lecturas:** Se espera que el sistema realice aproximadamente 500 a 1000 IOPS durante operaciones normales, aumentando a 2000 IOPS en momentos de alto tráfico.
 - **Escrituras:** En promedio, 200 a 500 IOPS para mantener el rendimiento adecuado en la escritura de transacciones y actualizaciones de saldo.

Diseño de Bajo Nivel

El diseño de bajo nivel describe la estructura detallada de los componentes del sistema. A continuación se presentan los módulos clave que manejan las operaciones bancarias del **Banco JP Morgan**, así como la lógica específica asociada a cada rol:

1. Pasarela de Pago:

- **Descripción:** Encargada de procesar los pagos realizados por los clientes. Incluye la validación de los fondos disponibles, la actualización de saldos y el manejo de posibles errores o reintentos.
- **Operaciones:** Validar transacciones, actualizar saldos, registrar el historial de transacciones, manejar pagos y reembolsos.

2. Servicio de Autenticación:

- **Descripción:** Gestiona la autenticación de los usuarios, asegurándose de que solo las personas autorizadas puedan acceder al sistema.
- **Operaciones:** Verificar credenciales, gestionar sesiones de usuario y controlar accesos.

3. Servicio de Préstamos:

- **Descripción:** Procesa las solicitudes de préstamos, realiza cálculos de tasas de interés y administra los pagos de los préstamos a lo largo del tiempo.
- **Operaciones:** Procesar solicitudes de préstamos, calcular pagos y plazos, verificar historial crediticio, aprobar o rechazar préstamos.

4. Servicio de Tarjetas de Crédito:

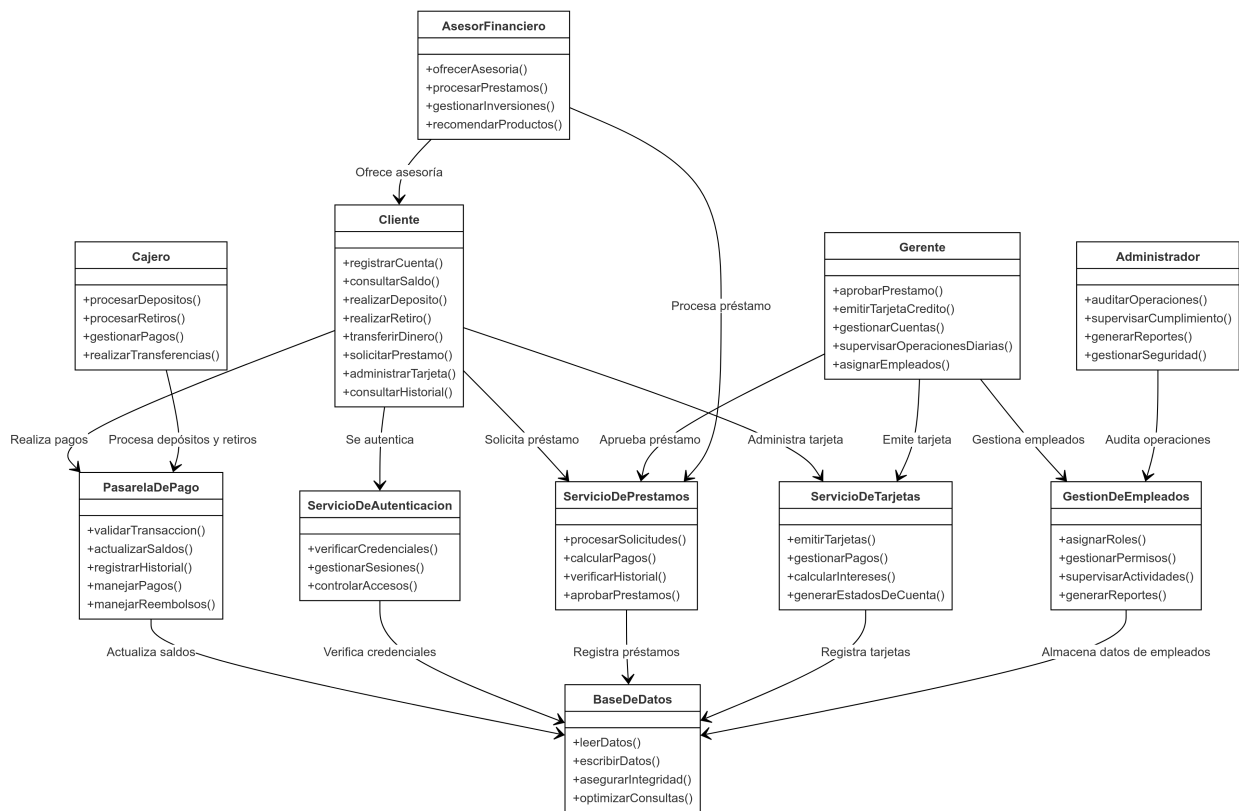
- **Descripción:** Gestiona la emisión y administración de tarjetas de crédito, incluyendo la verificación de límites de crédito, los ciclos de facturación y los pagos de las tarjetas.
- **Operaciones:** Emitir tarjetas, gestionar pagos de saldo, calcular intereses, generar estados de cuenta.

5. Módulo de Gestión de Empleados:

- **Descripción:** Maneja las operaciones relacionadas con los empleados del banco, desde la asignación de roles hasta la administración de sus cuentas y el control de accesos.
- **Operaciones:** Asignar roles a empleados, gestionar permisos de acceso, supervisar actividades y generar reportes.

6. Base de Datos:

- **Descripción:** La base de datos centralizada almacena toda la información del sistema, incluidas las cuentas de los clientes, transacciones, préstamos, tarjetas de crédito, y detalles de los empleados y sucursales.
- **Operaciones:** Lectura y escritura de datos, asegurar integridad de la información, manejo de índices para optimización de consultas.



Explicación del Diagrama:

1. **Cliente:** Interactúa con diferentes servicios del banco como la pasarela de pago, el servicio de préstamos, el servicio de tarjetas y el servicio de

autenticación para realizar transacciones, administrar sus cuentas y solicitar productos financieros.

2. **Gerente de Sucursal:** Gestiona la aprobación de préstamos, la emisión de tarjetas de crédito y la asignación de roles y responsabilidades a los empleados bajo su supervisión.
3. **Asesor Financiero:** Proporciona asesoría financiera a los clientes, procesa solicitudes de préstamos y gestiona inversiones y seguros para los clientes.
4. **Cajero:** Procesa depósitos, retiros y transferencias a través de la pasarela de pago.
5. **Administrador:** Realiza auditorías del sistema, asegura el cumplimiento normativo y gestiona la seguridad de todo el sistema.
6. **Pasarela de Pago:** Se encarga de validar las transacciones, actualizar los saldos y manejar cualquier error o reintento en el proceso de pagos.
7. **Servicio de Autenticación:** Verifica las credenciales y gestiona las sesiones y accesos de los usuarios.
8. **Servicio de Préstamos:** Procesa las solicitudes de préstamos y las mantiene en la base de datos, además de gestionar pagos e historial crediticio.
9. **Servicio de Tarjetas:** Administra la emisión y gestión de tarjetas de crédito, incluyendo pagos y generación de estados de cuenta.
10. **Gestión de Empleados:** Asigna roles y permisos a los empleados, y genera reportes de actividades.
11. **Base de Datos:** La base de datos centraliza toda la información del sistema, asegurando la integridad y optimización de las consultas para mejorar el rendimiento.

Diseño de Alto Nivel

El diseño de alto nivel se enfoca en la estructura general del sistema bancario del **Banco JP Morgan**, utilizando una arquitectura monolítica. En este enfoque, todos los componentes del sistema están integrados en una única aplicación, compartiendo una base de datos centralizada.

Descripción de los Componentes Principales:

1. Interfaz de Usuario (UI) esta solo se menciona pero no se debe de hacer:

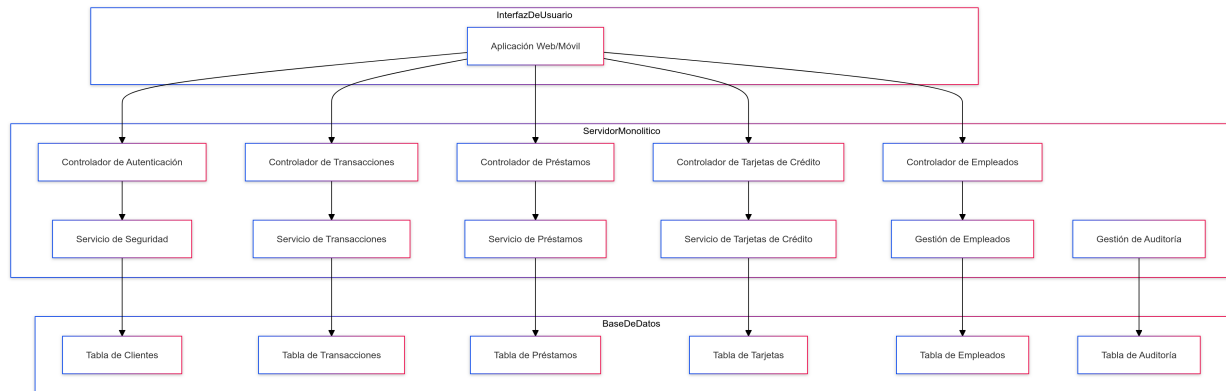
- La interfaz de usuario es el punto de contacto entre el cliente y el sistema. Puede ser una aplicación web o móvil desde donde los clientes y empleados interactúan con el sistema para realizar operaciones bancarias.

2. Capa de Aplicación (Servidor Monolítico):

- **Controladores:** Los controladores son responsables de manejar las solicitudes de la interfaz de usuario. Cada controlador está especializado en una parte del sistema, como la gestión de usuarios, transacciones, préstamos, tarjetas de crédito, y la gestión de empleados.
- **Servicios de Negocio:** Los servicios contienen la lógica del negocio que respalda las operaciones bancarias. Estos servicios procesan las solicitudes y garantizan que se respeten las reglas de negocio antes de interactuar con la base de datos.
- **Gestión de Seguridad:** Módulo encargado de manejar la autenticación y autorización de los usuarios, asegurando que solo los usuarios autorizados puedan realizar determinadas operaciones.
- **Gestión de Transacciones:** Módulo encargado de asegurar que todas las transacciones financieras se manejen de manera atómica, consistente, aislada y duradera (ACID).

3. Base de Datos Centralizada:

- La base de datos almacena toda la información del sistema, incluyendo datos de clientes, empleados, transacciones, préstamos y tarjetas de crédito. La base de datos está diseñada para manejar un alto volumen de operaciones y garantizar la integridad de los datos.



Explicación del Diagrama:

1. Interfaz de Usuario (UI):

- La interfaz de usuario (web o móvil) es utilizada por clientes y empleados para interactuar con el sistema bancario. Esta interfaz envía solicitudes a los controladores del servidor monolítico para realizar operaciones bancarias.

2. Servidor Monolítico:

- **Controladores:** Cada controlador maneja una parte específica del sistema:
 - **Controlador de Autenticación:** Gestiona el inicio de sesión y la autenticación de usuarios.
 - **Controlador de Transacciones:** Procesa las solicitudes de transacciones financieras, como depósitos, retiros y transferencias.
 - **Controlador de Préstamos:** Gestiona las solicitudes de préstamos, incluyendo la aprobación y el seguimiento de los pagos.
 - **Controlador de Tarjetas de Crédito:** Administra la emisión y gestión de tarjetas de crédito.
 - **Controlador de Empleados:** Gestiona la información relacionada con los empleados del banco.
- **Servicios de Negocio:** Los servicios son responsables de implementar la lógica empresarial detrás de las operaciones bancarias. Aseguran que las operaciones se realicen correctamente y que se respeten las reglas del negocio antes de interactuar con la base de datos.

- **Servicio de Seguridad:** Asegura que solo los usuarios autenticados y autorizados puedan realizar operaciones sensibles.
- **Servicio de Transacciones:** Gestiona las transacciones bancarias asegurando que se ejecuten bajo principios de consistencia y aislamiento.
- **Servicio de Préstamos:** Maneja todo el ciclo de vida de los préstamos, desde la solicitud hasta el pago.
- **Servicio de Tarjetas de Crédito:** Administra la emisión, uso y pagos de tarjetas de crédito.
- **Gestión de Empleados:** Se encarga de los roles y actividades de los empleados dentro del sistema bancario.
- **Gestión de Auditoría:** Supervisa y registra todas las actividades críticas del sistema para auditoría y monitoreo.

3. Base de Datos Centralizada:

- La base de datos almacena la información esencial del sistema. Está estructurada en tablas que contienen los datos de **clientes, transacciones, préstamos, tarjetas de crédito, empleados, y auditoría**. Todos los controladores y servicios acceden a esta base de datos centralizada para leer y escribir información.

Características de la Arquitectura Monolítica:

- **Simplicidad Inicial:** Al ser una aplicación monolítica, todos los componentes están integrados en una única aplicación, lo que simplifica el desarrollo, la implementación y el mantenimiento inicial.
- **Acceso Directo a la Base de Datos:** Todos los componentes del sistema interactúan directamente con la base de datos centralizada, lo que facilita las consultas y la gestión de datos.
- **Escalabilidad Vertical:** La arquitectura monolítica permite escalar la aplicación aumentando los recursos del servidor (CPU, memoria) para manejar un mayor volumen de operaciones.

- **Integración Completa:** La arquitectura monolítica ofrece una integración completa y fluida entre todos los módulos del sistema, reduciendo la complejidad de las comunicaciones entre ellos.

Consultas a la Base de Datos

El sistema bancario del **Banco JP Morgan** requiere realizar diversas consultas a la base de datos para gestionar las operaciones financieras de los clientes, préstamos, tarjetas de crédito, empleados, y auditoría. A continuación, se describen las principales consultas que se deben realizar en cada componente clave del sistema.

1. Consultas Relacionadas con Clientes

- **Registrar Cliente:** Insertar la información de un nuevo cliente en la base de datos cuando este abre una cuenta en el banco.
- **Actualizar Información del Cliente:** Modificar los datos de contacto y personales de un cliente existente.
- **Consultar Saldo del Cliente:** Obtener el saldo actual de una cuenta específica de un cliente.
- **Obtener Información del Cliente:** Recuperar los datos generales de un cliente, incluyendo nombre, dirección, teléfono y tipo de cuenta.

2. Consultas Relacionadas con Transacciones

- **Registrar Depósito:** Registrar un depósito realizado por un cliente en su cuenta bancaria.
- **Registrar Retiro:** Registrar un retiro realizado por un cliente en su cuenta bancaria.
- **Registrar Transferencia entre Cuentas:** Registrar una transferencia de fondos entre dos cuentas bancarias diferentes.
- **Consultar Historial de Transacciones del Cliente:** Recuperar el historial de todas las transacciones realizadas por un cliente en una cuenta específica.

3. Consultas Relacionadas con Préstamos

- **Solicitar Préstamo:** Registrar una nueva solicitud de préstamo de un cliente en el sistema.
- **Actualizar Estado del Préstamo:** Cambiar el estado de un préstamo (aprobado, rechazado, pendiente) tras la evaluación del mismo.
- **Registrar Pago de Préstamo:** Registrar un pago realizado por un cliente hacia su préstamo.
- **Consultar Préstamos Activos del Cliente:** Listar los préstamos aprobados y activos de un cliente, incluyendo el saldo pendiente y las condiciones del préstamo.

4. Consultas Relacionadas con Tarjetas de Crédito

- **Emitir Tarjeta de Crédito:** Registrar la emisión de una nueva tarjeta de crédito a un cliente.
- **Registrar Pago de Tarjeta de Crédito:** Registrar un pago realizado por el cliente para cubrir el saldo de su tarjeta de crédito.
- **Registrar Compra con Tarjeta de Crédito:** Registrar una compra realizada con la tarjeta de crédito de un cliente.
- **Consultar Estado de Cuenta de Tarjeta de Crédito:** Recuperar el estado de cuenta de la tarjeta de crédito de un cliente, incluyendo saldo, límite y fecha de corte.

5. Consultas Relacionadas con Empleados

- **Registrar Nuevo Empleado:** Insertar la información de un nuevo empleado en la base de datos.
- **Actualizar Información del Empleado:** Modificar los datos personales o salariales de un empleado existente.
- **Consultar Empleados de una Sucursal:** Listar todos los empleados que trabajan en una sucursal o agencia específica.

6. Consultas Relacionadas con Sucursales y Agencias

- **Registrar Nueva Sucursal/Agencia:** Insertar los datos de una nueva sucursal o agencia del banco.

- **Consultar Sucursales por Departamento:** Recuperar una lista de todas las sucursales o agencias ubicadas en un departamento específico.

7. Consultas de Auditoría y Reportes

- **Auditoría de Transacciones:** Realizar auditorías de las transacciones realizadas en el sistema en un período determinado.
- **Generar Reporte de Préstamos Activos:** Generar reportes que muestren todos los préstamos activos en el banco, con detalles como saldo pendiente y estado.

8. Consultas de Seguridad

- **Verificar Credenciales de Usuario:** Comprobar las credenciales de acceso de los usuarios que intentan autenticarse en el sistema.
- **Control de Accesos:** Consultar los permisos y roles asignados a un usuario para controlar qué operaciones pueden realizar en el sistema.

Implementación de Código API

1. API de Registro de Usuarios (POST)

- **Extremo:** `/api/user/register`
- **Descripción:** Permite a los usuarios crear sus cuentas de forma segura. Recibe los datos del usuario y los almacena en la base de datos.

Solicitud:

```
{
  "username": "example_user",
  "email": "user@example.com",
  "password": "securepassword123"
}
```

Respuesta:

```
{
  "status": "success",
}
```

```
{
  "message": "User registration successful"
}
```

2. API de Autenticación de Usuarios (POST)

- **Extremo:** `/api/user/login`
- **Descripción:** Permite a los usuarios autenticarse utilizando su nombre de usuario y contraseña.

Solicitud:

```
{
  "username": "example_user",
  "password": "securepassword123"
}
```

Respuesta:

```
{
  "status": "success",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

3. API de Consulta de Saldo (GET)

- **Extremo:** `/api/account/balance`
- **Descripción:** Permite a un usuario consultar el saldo de su cuenta bancaria. Requiere autenticación.

Solicitud (Cabecera de Autenticación JWT):

```
Authorization: Bearer <token>
```

Respuesta:

```
{
  "status": "success",
  "balance": 1500.00
}
```

4. API de Depósito (POST)

- **Extremo:** `/api/transaction/deposit`
- **Descripción:** Permite a un cliente realizar un depósito en su cuenta bancaria.

Solicitud:

```
{
  "account_id": "5001",
  "amount": 1000.00
}
```

Respuesta:

```
{
  "status": "success",
  "message": "Deposit successful",
  "new_balance": 2500.00
}
```

5. API de Retiro (POST)

- **Extremo:** `/api/transaction/withdraw`
- **Descripción:** Permite a un cliente realizar un retiro de su cuenta bancaria.

Solicitud:

```
{
  "account_id": "5001",
```

```
"amount": 500.00
}
```

Respuesta:

```
{
  "status": "success",
  "message": "Withdrawal successful",
  "new_balance": 2000.00
}
```

6. API de Transferencia entre Cuentas (POST)

- **Extremo:** `/api/transaction/transfer`
- **Descripción:** Permite realizar una transferencia entre dos cuentas del banco.

Solicitud:

```
{
  "from_account_id": "5001",
  "to_account_id": "5002",
  "amount": 1000.00
}
```

Respuesta:

```
{
  "status": "success",
  "message": "Transfer successful",
  "from_account_balance": 1000.00,
  "to_account_balance": 3000.00
}
```

7. API de Registro de Préstamo (POST)

- **Extremo:** `/api/loan/apply`

- **Descripción:** Permite a un cliente solicitar un préstamo.

Solicitud:

```
{
  "client_id": "1001",
  "amount": 50000.00,
  "interest_rate": 0.05,
  "term_months": 24
}
```

Respuesta:

```
{
  "status": "success",
  "message": "Loan application successful",
  "loan_id": "3001"
}
```

8. API de Pago de Préstamo (POST)

- **Extremo:** `/api/loan/payment`
- **Descripción:** Permite a un cliente realizar un pago a su préstamo.

Solicitud:

```
{
  "loan_id": "3001",
  "payment_amount": 2000.00
}
```

Respuesta:

```
{
  "status": "success",
  "message": "Loan payment successful",
}
```

```
"remaining_balance": 48000.00
}
```

9. API de Consulta de Préstamos (GET)

- **Extremo:** `/api/loan/list`
- **Descripción:** Permite a un cliente consultar sus préstamos activos.

Solicitud (Cabecera de Autenticación JWT):

```
Authorization: Bearer <token>
```

Respuesta:

```
{
  "status": "success",
  "loans": [
    {
      "loan_id": "3001",
      "amount": 50000.00,
      "remaining_balance": 48000.00,
      "interest_rate": 0.05,
      "term_months": 24
    }
  ]
}
```

10. API de Consulta de Historial de Transacciones (GET)

- **Extremo:** `/api/transaction/history`
- **Descripción:** Permite a un cliente consultar el historial de transacciones de su cuenta.

Solicitud (Cabecera de Autenticación JWT):

```
Authorization: Bearer <token>
```


Respuesta:

```
{
  "status": "success",
  "transactions": [
    {
      "transaction_id": "12345",
      "type": "Deposit",
      "amount": 1000.00,
      "date": "2024-08-14T12:00:00Z"
    },
    {
      "transaction_id": "12346",
      "type": "Withdrawal",
      "amount": 500.00,
      "date": "2024-08-15T14:00:00Z"
    }
  ]
}
```

Datos Disponibles

Los siguientes archivos CSV están disponibles para cargar datos en la base de datos:

1. Clientes (`clientes_banco_jp_morgan_1m.csv`)

Campos:

1. **ID:** Identificador único del cliente.
2. **Nombre:** Primer nombre del cliente.
3. **Apellido:** Apellido del cliente.
4. **Número de Cuenta:** Número de cuenta bancaria del cliente.
5. **Tipo de Cuenta:** Clasificación del tipo de cuenta bancaria (e.g., corriente, ahorro).

- 6. **Saldo:** Monto actual en la cuenta.
- 7. **Teléfono:** Número de contacto del client

2. Transacciones (`transacciones_banco_jp_morgan_500k.csv`)

Campos:

- 1. **ID Transacción:** Identificador único de la transacción.
- 2. **ID Cliente:** Identificador único del cliente que realizó la transacción.
- 3. **Número de Cuenta:** Número de cuenta del cliente.
- 4. **Tipo de Transacción:** Tipo de transacción realizada (Depósito, Retiro, Transferencia, Pago).
- 5. **Monto:** Monto de la transacción.
- 6. **Fecha:** Fecha en que se realizó la transacción.
- 7. **Hora:** Hora en que se realizó la transacción.
- 8. **Descripción:** Breve descripción de la transacción.
- 9. **Sucursal/Agencia:** Sucursal o agencia donde se realizó la transacción.

3. Préstamos (`prestamos_banco_jp_morgan_100k.csv`)

Campos a Incluir:

- 1. **ID Préstamo:** Identificador único del préstamo.
- 2. **ID Cliente:** Identificador único del cliente que recibió el préstamo.
- 3. **Monto del Préstamo:** Monto total del préstamo otorgado.
- 4. **Tasa de Interés:** Tasa de interés aplicada al préstamo.
- 5. **Fecha de Desembolso:** Fecha en que se desembolsó el préstamo.
- 6. **Fecha de Vencimiento:** Fecha en que se debe liquidar el préstamo.
- 7. **Saldo Pendiente:** Saldo que aún está pendiente de pago.
- 8. **Estado del Préstamo:** Estado actual del préstamo (Activo, Vencido, Cancelado).

4. Tarjetas de Crédito (`tarjetas_credito_banco_jp_morgan_100k.csv`)

Campos:

1. **ID Tarjeta:** Identificador único de la tarjeta de crédito.
 2. **ID Cliente:** Identificador único del cliente que posee la tarjeta.
 3. **Número de Tarjeta:** Número de la tarjeta de crédito.
 4. **Límite de Crédito:** Límite máximo de crédito asignado a la tarjeta.
 5. **Saldo Actual:** Saldo actual pendiente en la tarjeta.
 6. **Fecha de Emisión:** Fecha en que se emitió la tarjeta.
 7. **Fecha de Expiración:** Fecha en que expira la tarjeta.
 8. **Estado:** Estado actual de la tarjeta (Activa, Bloqueada, Cancelada).
 9. **Fecha de Corte:** La fecha en que se genera el estado de cuenta de la tarjeta de crédito.
 10. **Día del Ciclo:** El día del mes en que se cierra el ciclo de facturación de la tarjeta de crédito (puede variar entre 1 y 31).
5. **Sucursales/Agencias** ([sucursales_agencias_banco_jp_morgan.csv](#))

Campos:

1. **ID:** Identificador único de la sucursal o agencia.
2. **Nombre:** Nombre de la sucursal o agencia.
3. **Tipo:** Especifica si es una sucursal o una agencia.
4. **Departamento:** Departamento en el que se encuentra la sucursal o agencia.
5. **Municipio:** Municipio correspondiente.
6. **Dirección:** Dirección específica de la sucursal o agencia.
7. **Código Postal:** Código postal del área.
8. **Teléfono:** Número de contacto de la sucursal o agencia.
9. **Horario de Atención:** Horario de atención al cliente.
10. **Número de Empleados:** Número de empleados en la sucursal o agencia.

11. **Servicios Disponibles:** Servicios ofrecidos en la sucursal o agencia (Cajero automático, Préstamos, Asesoría Financiera, etc.).

6. Empleados (`empleados_banco_jp_morgan.csv`)

Campos:

1. **ID:** Identificador único del empleado.
2. **Nombre:** Nombre del empleado.
3. **Apellido:** Apellido del empleado.
4. **Rol:** El rol o posición del empleado (Ej. Gerente de Sucursal, Asesor Financiero, Cajero, etc.).
5. **Departamento:** Departamento al que está asignado el empleado.
6. **Sucursal/Asignación:** La sucursal o agencia a la que está asignado el empleado.
7. **Teléfono:** Número de contacto del empleado.
8. **Fecha de Contratación:** La fecha en que el empleado fue contratado.
9. **Salario:** El salario del empleado.
10. **Horario de Trabajo:** Horario en el que el empleado trabaja.

Entregables

Cada estudiante tendrá que estructurar su repositorio de manera organizada y documentada para facilitar la revisión del proyecto.

1. Estructura del Repositorio en GitHub

- Los estudiantes deberán crear un repositorio en GitHub con el nombre `Proyecto1_SistemaBancario_<carnet_estudiante>`.
- Dentro del repositorio, deberán organizar el proyecto en carpetas adecuadas para cada componente, por ejemplo:
 - `/sql/`: Scripts SQL para crear tablas, índices, y consultas.

- `/src/` : Código fuente del sistema bancario Python o Node.js.
- `/docs/` : Documentación del proyecto, incluyendo diagramas, manual de usuario y explicación del diseño.
- `/tests/` : Casos de prueba (las consultas) y scripts de pruebas unitarias o de integración.

2. Documentación

- **README.md**: Debe incluir una descripción del proyecto, instrucciones de instalación y configuración, así como ejemplos de uso.
- **Explicación del Diseño**: Un documento que describa el diseño de alto y bajo nivel, incluyendo los componentes principales del sistema y cómo interactúan entre sí.
- **Explicación de Consultas a la Base de Datos**: Documento que explique las consultas realizadas a la base de datos, sin incluir el código SQL (esto se entregará por separado en los scripts SQL).

3. Código Fuente del Proyecto

- **Implementación del Sistema**: El código del sistema bancario debe estar estructurado de manera adecuada. Esto incluye la implementación de los controladores, servicios y la lógica de negocio definida en el proyecto.
- **Scripts SQL**: Scripts para la creación de las tablas de la base de datos, índices.
- **Consultas SQL**: Los estudiantes deberán incluir las consultas SQL que se utilizarán en el sistema, como las que se utilizan para gestionar clientes, transacciones, préstamos, tarjetas de crédito y empleados (las consultas que se mencionan anteriormente).

4. Pruebas

- **Pruebas Unitarias de cada una de las consultas**: Deben incluir pruebas unitarias de las funciones clave del sistema, como la autenticación de usuarios, el procesamiento de transacciones y la gestión de préstamos y tarjetas de crédito.

6. Entrega y Control de Versiones

- **Commit Inicial:** El commit inicial debe incluir la estructura básica del proyecto y un README que explique los objetivos del mismo.
- **Commits Frecuentes:** Los estudiantes deben realizar commits frecuentes, cada uno de ellos acompañado de mensajes claros que expliquen los cambios realizados. Esto es esencial para demostrar un desarrollo iterativo y controlado.

Resumen de Entregables

- **Repositorio en GitHub** con toda la estructura del proyecto entrega en UEDI deben de agregar al tutor usuario de GitHub OrmandyRony.
- **Documentación completa** del diseño, diagramas y explicación de consultas.
- **Código fuente** del sistema bancario.
- **Scripts SQL** y consultas.
- **Pruebas unitarias**

Herramientas

- Base de datos Oracle Dockerizada.
- Data modeler o Lucihcart (No se debe crear automáticamente el script para hacer la base de datos).
- Se recomienda usar Node.js en ello se darán ejemplos pero el lenguaje de api.
- Se recomienda usar Python para la carga de datos o una aplicación de software cliente de SQL.