

**Universidad Peruana de Ciencias Aplicadas**

**Facultad de Ingeniería**

Fundamentos de Sistemas de Información

**Trabajo Final**

**Nombre de la Empresa: Caja Arequipa**

**Docente:** Orestes Manolo Ramirez Garcia

**Sección:** SS51

**Integrantes:**

* + Caballero Balladares, Niels Ygdaly (u202120268)
  + Farroñan Cruz, Aldair (u20211D742)
  + Jara Figueroa, Nemberk Jesús (u202120739)
  + Pozo Coras, Keo Franck Luis (u202120644)
  + Vivanco Chavez, Elian Salvador (u201922586)

**Semestre 2023-2**

**Índice:**

1.Objetivo del Estudiante ………………………………………..……………………………

2.Descripción de la empresa …………………………………..……………………………… 3.Descripción del área ...………………………………………….……………………………. 4.Descripción de los procesos y las mejoras a aplicar ……….……………….. ………………. 5.Requerimientos Funcionales, no Funcionales y mockps………………………………………

6.Diagrama de base de datos relacional y diagrama de clases ………………………………...

7.Database Code …………………………………………………………………………….….

8. Scripts to Relational Database………………………………………………………………..

9. Manual de usuario……………………………………………………………………………

10. Conclusiones y recomendaciones …………………………………………………………

10.Anexos …………………………………………………………………….………………..

11.Bibliografía …………………………………………………….…………………………..

# I. Objetivo del Estudiante

|  |  |  |  |
| --- | --- | --- | --- |
| Student Outcome ABET Criterios | | Criterios | Actividades individuales |
| ABET 5 | La capacidad de funcionar efectivamente como miembro o líder en un equipo cuyos miembros juntos proporcionan liderazgo, crean un entorno de colaboración e inclusivo, establecen objetivos, planifican tareas y cumplen objetivos | Puntaje especifico | 20 |
| 5.c1. Participa en equipos multidisciplinarios con eficacia, eficiencia y objetividad, en el marco de un proyecto en soluciones de sistemas de información | Todos los integrantes asistieron regularmente a las reuniones programadas del equipo, mostrando compromiso con el proyecto.  Cada miembro contribuye de manera significativa a las discusiones, aportando ideas constructivas y participando en la toma de decisiones del grupo |
| Puntaje especifico | 20 |
| 5.c2. Conoce al menos un sector empresarial o dominio de aplicación de soluciones de tecnologías de la información. | Todos los integrantes investigaron a fondo un sector empresarial vinculado a soluciones de tecnologías de la información.  Cada miembro participo de manera activa en las discusiones grupales, compartiendo hallazgos y perspectivas para enriquecer la comprensión colectiva del equipo sobre las implicaciones tecnológicas en el sector. |
| Puntaje especifico | 20 |
| 5.c3. Conocimientos de nuevos métodos de colaboración y comunicación | Cada miembro del grupo investigo herramientas de colaboración moderna, como Microsoft Teams, o Trello. |

# PRESENTACIÓN

# Descripción de la empresa

Caja Arequipa es una institución financiera de carácter cooperativo y orientada al bienestar económico de sus asociados. Con una larga trayectoria en el mercado financiero, Caja Arequipa se ha destacado por su compromiso en brindar servicios financieros de calidad a sus clientes.

# Descripción del área

**Área: Financiera**

Caja Arequipa se especializa en ofrecer una amplia gama de productos y servicios financieros que incluyen cuentas de ahorro, cuentas corrientes, préstamos personales, hipotecas y asesoría financiera. Su enfoque principal es proporcionar soluciones financieras que ayuden a sus clientes a alcanzar sus metas y objetivos económicos.

Asimismo, se caracteriza por su compromiso con la población, promoviendo el ahorro responsable y el acceso a servicios financieros inclusivos. A través de una red de sucursales y servicios en línea, la institución trabaja para ofrecer soluciones financieras confiables y eficientes a sus clientes, contribuyendo así al desarrollo económico de cada región que opera.

# Descripción de los procesos y las mejoras a aplicar

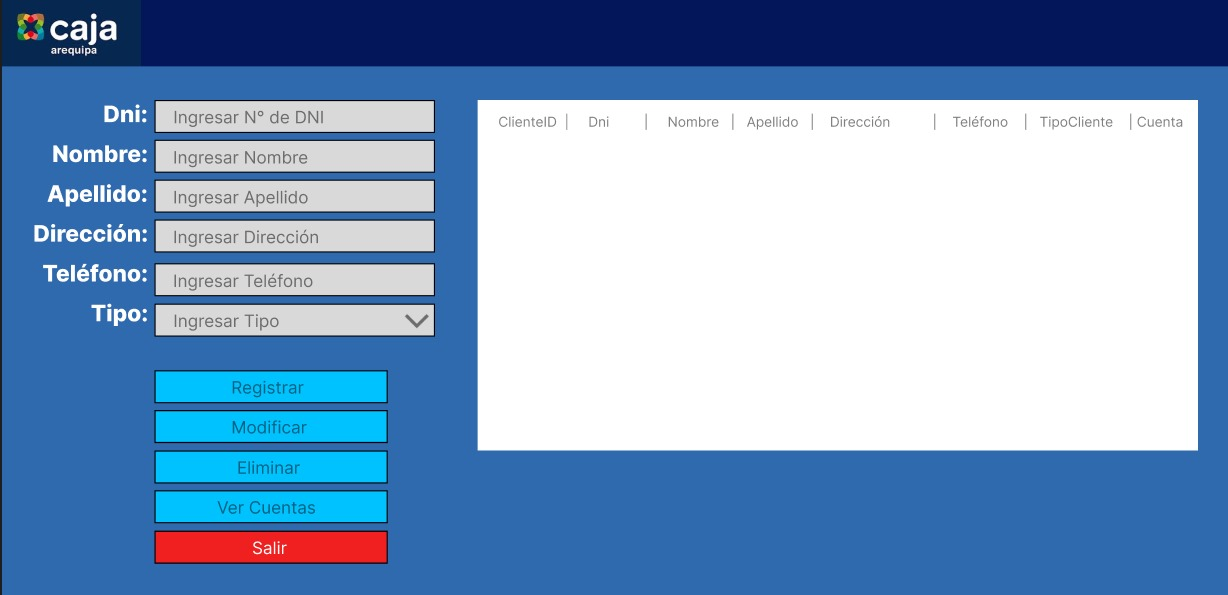
* **Proceso de Solicitud de Préstamos:** Actualmente, el proceso de solicitud de préstamos puede ser largo y requerir una cantidad significativa de documentación. La aplicación podría simplificar este proceso permitiendo a los clientes cargar documentos de manera electrónica y seguir el estado de su solicitud en tiempo real. Además, se podrían implementar algoritmos de análisis de riesgos para agilizar la aprobación de préstamos.
* **Gestión de Información de Clientes**: Caja Arequipa necesita mantener un registro actualizado de la información de sus clientes. La aplicación podría automatizar la recopilación y actualización de datos, incluyendo información personal, estados de cuenta, historial crediticio, entre otros. Esto ayudaría a la entidad a brindar un mejor servicio al cliente y tomar decisiones informadas sobre la concesión de préstamos.

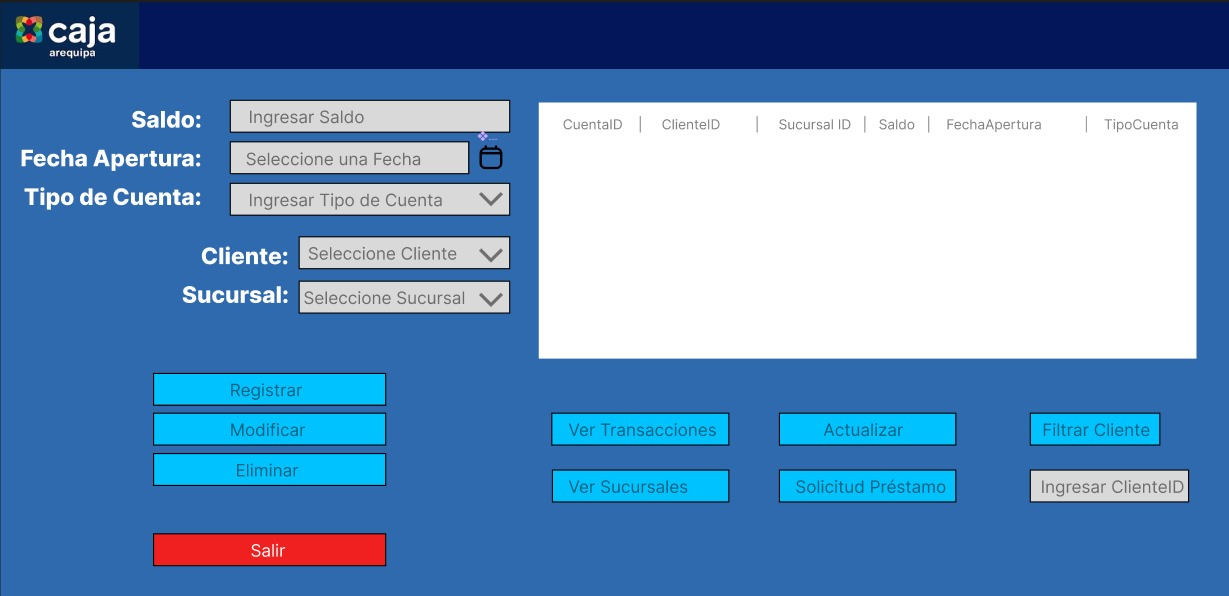
# Requerimientos Funcionales y no Funcionales

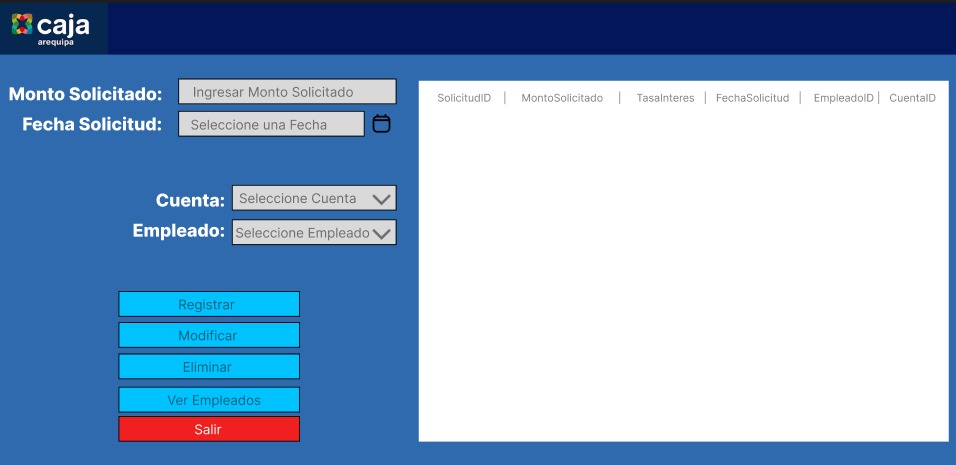
* **Requerimientos Funcionales:**
  + **Registro de Clientes:** El sistema debe permitir la creación y mantenimiento de registros de clientes, incluyendo información personal, datos de contacto y detalles de identificación.
  + **Gestión de Préstamos:** Debe ser posible registrar y administrar solicitudes de préstamos, incluyendo la aprobación, seguimiento y rechazo de solicitudes. Además, el sistema debe calcular automáticamente los plazos y tasas de interés.
  + **Generación de Contratos:** El sistema debe generar automáticamente contratos de préstamo con todos los términos y condiciones necesarios, que luego pueden ser impresos o enviados electrónicamente a los clientes.
  + **Seguimiento de Pagos:** Debe permitir el seguimiento de los pagos de los clientes, incluyendo el registro de pagos parciales y el cálculo de saldos pendientes.
  + **Generación de Informes:** Debe ser capaz de generar informes financieros y de seguimiento, como estados de cuenta, resumen de préstamos y análisis de riesgo.
  + **Acceso de Usuarios:** El sistema debe proporcionar diferentes niveles de acceso a los usuarios, como administradores, agentes de préstamos y clientes, con roles y permisos específicos.
* **Requerimientos No funcionales:**
  + **Seguridad de Datos:** El sistema debe cumplir con altos estándares de seguridad para proteger la información financiera y personal de los clientes.
  + **Rendimiento:** El sistema debe ser capaz de manejar grandes volúmenes de datos y transacciones sin degradación significativa del rendimiento.
  + **Disponibilidad:** Debe estar disponible las 24 horas del día, los 7 días de la semana, con tiempos de inactividad programados mínimos para mantenimiento.
  + **Usabilidad:** La interfaz de usuario debe ser intuitiva y fácil de usar para que los empleados de Caja Arequipa puedan aprender a utilizar el sistema con rapidez.
  + **Escalabilidad:** El sistema debe ser escalable para poder adaptarse a futuras expansiones y cambios en la cantidad de clientes y transacciones.
  + **Documentación:** Debe proporcionarse una documentación completa y actualizada para usuarios y administradores, incluyendo manuales de usuario y guías de mantenimiento.

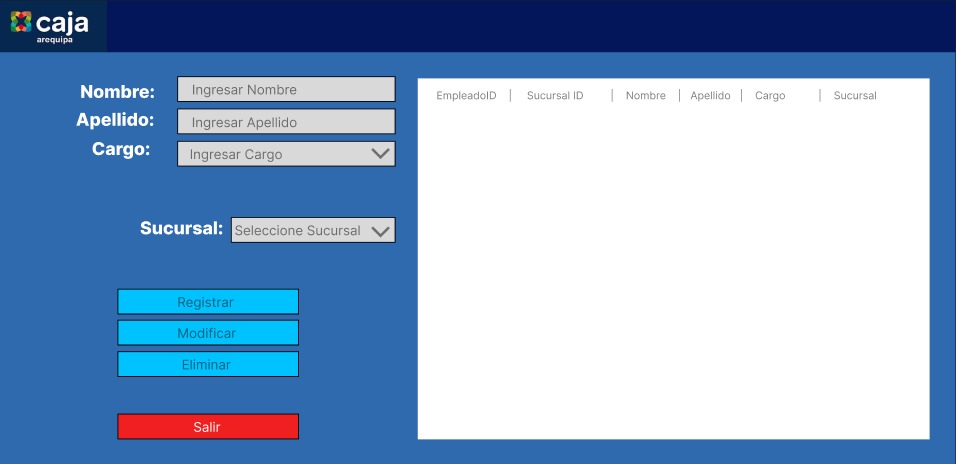
Mockups:

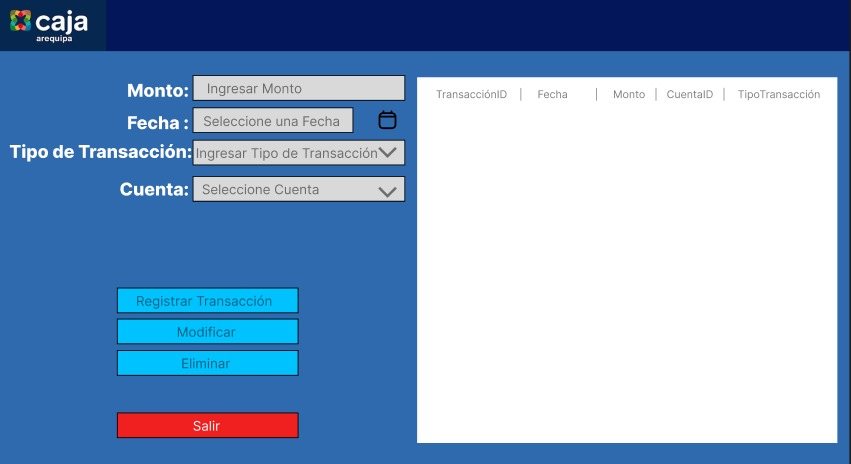


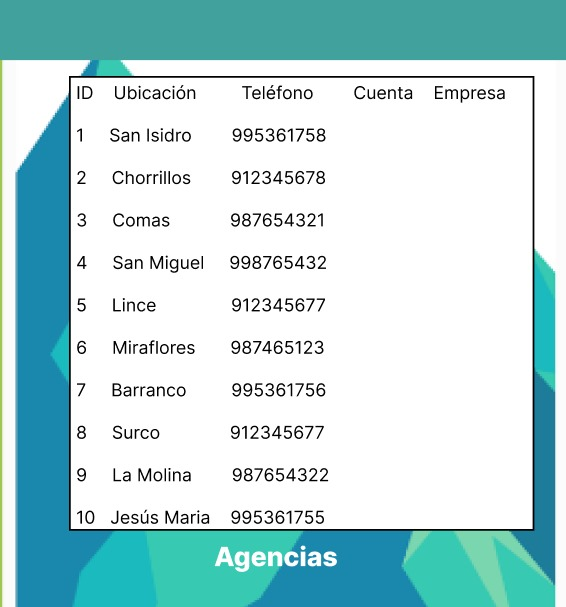












1. **DIAGRAMA DE CLASES:**

Diagrama de Clases

Descripción generada automáticamente

# Diagrama de base de datos relacional

Diagrama

Descripción generada automáticamente

En este punto explicaremos cada tabla impuesta en el modelado de Vertabelo.

**TABLA CLIENTE**

La tabla "Cliente" permite registrar un cliente de la empresa, el cual debe contener un identificador como el DNI al momento de registrarse. Adicionalmente, se deben registrar los datos adicionales correspondientes, como el nombre, apellido, dirección, teléfono y su de tipo de cliente. El cliente tiene la posibilidad de solicitar una o más cuentas y realizar solicitudes de préstamo.

Tabla

Descripción generada automáticamente

**TABLA SOLICITUD PRESTAMO**

La tabla "Solicitud Préstamo" contendrá un propio PrestamoID, que servirá para identificarla al momento de acceder a la información de la cuenta. En cada préstamo se debe proporcionar el id de la cuenta, monto solicitado, tasa de interés y la fecha de solicitud.

Tabla

Descripción generada automáticamente

**TABLA EMPLEADO**

Luego tenemos la tabla "Empleado", que se identificará mediante un EmpleadoID. El empleado tendrá que ser registrado con un nombre y apellido. Además, se deberá especificar el cargo.

Tabla

Descripción generada automáticamente

**TABLA CUENTA**

La tabla 'Cuenta' almacena información sobre las cuentas de clientes en el sistema. Cada cuenta tiene su identificador único (CuentaID) para facilitar el acceso a los detalles de la cuenta. Se vincula a un cliente mediante ClienteID. Además, se registra el saldo actual de la cuenta y la fecha de apertura. Esta tabla es fundamental para gestionar las cuentas de los clientes y realizar un seguimiento de las transacciones financieras asociadas a cada una.

Tabla

Descripción generada automáticamente

**TABLA SUCURSAL**

La tabla "Sucursal" sirve como entidad de registro para las sucursales de nuestra empresa en el sistema. Cada sucursal se identifica mediante un ID único (clave primaria) y se almacena su Ubicación y teléfono. Esto permite organizar y gestionar de manera efectiva la información geográfica de las sucursales.

Tabla

Descripción generada automáticamente

**TABLA TRANSACCIONES**

La tabla "transaccion" sirve como entidad de registro para las transacciones de los clientes en el sistema. Cada registro se identifica mediante un ID único (clave primaria) y se almacena sus atributos respectivos como el monto y la fecha. Además, se relaciona con un tipo y con una cuenta de un cliente específico a través de sus claves foráneas. Esto permite organizar y gestionar de manera efectiva las operaciones de cada cliente.

Tabla

Descripción generada automáticamente

**Database code:**

Create database TF\_Fundamentos\_Sistemas\_Informacion

Use TF\_Fundamentos\_Sistemas\_Informacion

-- Table: Cliente

CREATE TABLE Cliente (

ClienteID int NOT NULL IDENTITY,

Dni nvarchar(8) NOT NULL,

Nombre nvarchar(20) NOT NULL,

Apellido nvarchar(20) NOT NULL,

Direccion nvarchar(50) NOT NULL,

Telefono nvarchar(9) NOT NULL,

Tipo\_Cliente nvarchar(50) NOT NULL,

CONSTRAINT Cliente\_pk PRIMARY KEY (ClienteID)

);

-- Table: Cuenta

CREATE TABLE Cuenta (

CuentaID int NOT NULL IDENTITY,

ClienteID int NOT NULL,

Sucursal\_ID int NOT NULL,

Saldo decimal(10,4) NOT NULL,

FechaApertura date NOT NULL,

Tipo\_Cuenta nvarchar(50) NOT NULL,

CONSTRAINT Cuenta\_pk PRIMARY KEY (CuentaID)

);

-- Table: Empleado

CREATE TABLE Empleado (

EmpleadoID int NOT NULL IDENTITY,

Sucursal\_ID int NOT NULL,

Nombre nvarchar(30) NOT NULL,

Apellido nvarchar(30) NOT NULL,

Cargo nvarchar(30) NOT NULL,

CONSTRAINT Empleado\_pk PRIMARY KEY (EmpleadoID)

);

-- Table: SolicitudPrestamo

CREATE TABLE SolicitudPrestamo (

SolicitudID int NOT NULL IDENTITY,

MontoSolicitado decimal(10,2) NOT NULL,

TasaInteres decimal(10,2) NOT NULL,

FechaSolicitud date NOT NULL,

EmpleadoID int NOT NULL,

CuentaID int NOT NULL,

CONSTRAINT SolicitudPrestamo\_pk PRIMARY KEY (SolicitudID)

);

-- Table: Sucursal

CREATE TABLE Sucursal (

ID int NOT NULL IDENTITY,

Ubicacion nvarchar(20) NOT NULL,

Telefono nvarchar(9) NOT NULL,

CONSTRAINT Sucursal\_pk PRIMARY KEY (ID)

);

-- Table: Transacciones

CREATE TABLE Transacciones (

TransaccionID int NOT NULL IDENTITY,

Fecha date NOT NULL,

Monto decimal(10,4) NOT NULL,

CuentaID int NOT NULL,

Tipo\_Transaccion nvarchar(50) NOT NULL,

CONSTRAINT Transacciones\_pk PRIMARY KEY (TransaccionID)

);

-- foreign keys

-- Reference: Cuenta\_Cliente (table: Cuenta)

ALTER TABLE Cuenta ADD CONSTRAINT Cuenta\_Cliente

FOREIGN KEY (ClienteID)

REFERENCES Cliente (ClienteID);

-- Reference: Cuenta\_Sucursal (table: Cuenta)

ALTER TABLE Cuenta ADD CONSTRAINT Cuenta\_Sucursal

FOREIGN KEY (Sucursal\_ID)

REFERENCES Sucursal (ID);

-- Reference: Empleado\_Sucursal (table: Empleado)

ALTER TABLE Empleado ADD CONSTRAINT Empleado\_Sucursal

FOREIGN KEY (Sucursal\_ID)

REFERENCES Sucursal (ID);

-- Reference: SolicitudPrestamo\_Cuenta (table: SolicitudPrestamo)

ALTER TABLE SolicitudPrestamo ADD CONSTRAINT SolicitudPrestamo\_Cuenta

FOREIGN KEY (CuentaID)

REFERENCES Cuenta (CuentaID);

-- Reference: SolicitudPrestamo\_Empleado (table: SolicitudPrestamo)

ALTER TABLE SolicitudPrestamo ADD CONSTRAINT SolicitudPrestamo\_Empleado

FOREIGN KEY (EmpleadoID)

REFERENCES Empleado (EmpleadoID);

-- Reference: Transacciones\_Cuenta (table: Transacciones)

ALTER TABLE Transacciones ADD CONSTRAINT Transacciones\_Cuenta

FOREIGN KEY (CuentaID)

REFERENCES Cuenta (CuentaID);

Scripts to Relational Database

INSERT INTO Cliente (Dni, Nombre, Apellido, Direccion, Telefono, Tipo\_Cliente)

VALUES

('61164069', 'Juan', 'Perez', 'San Miguel, Lima, Peru', '987654321', 'Empresarial'),

('71248783', 'Maria', 'Gomez', 'Comas, Lima, Peru', '987123456', 'Inversionista'),

('61123987', 'Carlos', 'Lopez', 'Chorrillos, Lima, Peru', '987987654', 'Pensionado'),

('71234567', 'Ana', 'Martinez', 'San Isidro, Lima, Peru', '987789012', 'Estudiante'),

('61198765', 'Laura', 'Rodriguez', 'Lince, Lima, Peru', '987345678', 'Empresarial');

INSERT INTO Sucursal (Ubicacion, Telefono)

VALUES

('San Isidro', '995361758'),

('Chorrillos', '912345678'),

('Comas', '987654321'),

('San Miguel', '998765432'),

('Lince', '912345678'),

('Miraflores', '987654321'),

('Barranco', '995361758'),

('Surco', '912345678'),

('La Molina', '987654321'),

('Jesús María', '995361758');

INSERT INTO Empleado (Sucursal\_ID, Nombre, Apellido, Cargo)

VALUES

(1, 'Carlos', 'Gutierrez', 'Gerente'),

(2, 'Ana', 'Rodriguez', 'Cajero'),

(3, 'Pedro', 'Lopez', 'Asesor Comercial'),

(4, 'María', 'Fernández', 'Cajero');

INSERT INTO Cuenta (ClienteID, Sucursal\_ID, Saldo, FechaApertura, Tipo\_Cuenta)

VALUES

(1, 1, 1500.00, '2023-11-16', 'Cuenta de Ahorro'),

(2, 2, 2500.00, '2023-11-16', 'Cuenta Corriente'),

(3, 3, 1800.00, '2023-11-16', 'Cuenta de Ahorro'),

(4, 4, 2000.00, '2023-11-16', 'Cuenta Corriente'),

(5, 5, 3000.00, '2023-11-16', 'Cuenta de Ahorro');

INSERT INTO SolicitudPrestamo (MontoSolicitado, TasaInteres, FechaSolicitud, EmpleadoID,CuentaID)

VALUES

(5000.00, 0.05, '2023-08-16', 1, 1),

(8000.00, 0.08, '2023-11-16', 3, 3),

(10000.00, 0.07, '2023-10-13', 4, 4),

(12000.00, 0.06, '2023-11-15', 2, 2),

(6000.00, 0.09, '2022-12-14', 5, 5);

INSERT INTO Transacciones (Fecha, Monto,CuentaID, Tipo\_Transaccion)

VALUES

('2023-11-16', 500.00, 1, 'Deposito'),

('2023-11-17', 200.00, 2, 'Retiro'),

('2023-11-18', 1000.00, 3, 'Transferencia'),

('2023-11-19', 150.00, 4, 'Deposito'),

('2023-11-20', 300.00, 5, 'Retiro');

# Manual del Usuario

PANTALLA PRINCIPAL:

Tenemos un formulario de nuestro programa que esta el botón de ver clientes y para salir de la venta.

Imagen que contiene Gráfico

Descripción generada automáticamente

CLIENTES:

Vemos aquí los registros de los clientes

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Luego podemos observar un ejemplo sobre la funcionalidad de “registrar” un cliente.

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Podemos ver un ejemplo de una modificacion, queremos cambiar los datos del cliente con el ID igual a 15.

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Ademas, cumplimos con la funcionalidad de eliminar un registro si es que se requiere. En este caso eliminaremos el Cliente con el ID 12.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

CUENTAS:

Ahora vamos con otra tabla de cuentas, en nuestra base de datos definimos que un cliente puede tener uno o varias cuentas por sucursal.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Vemos un ejemplo de un registro de una cuenta, se coloca los campos y seleccionamos el cliente que se asignara la cuenta y con la sucursal, en este caso es el cliente “Nemberk” de la sucursal “Comas”.

Se guardara los datos insertados pero los campos de cliente y sucursal seran colocados con su respectivo ID.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Vamos con la funcionalidad de Filtrar las cuentas por Cliente. Lo definimos como una relacion de una muchos en la base de datos. Observamos un ejemplo de una filtracion del cliente con el ID igual a 5, que se llama Laura Rodriguez y vemos que tiene dos cuentas, de ahorro y una corriente.

Interfaz de usuario gráfica

Descripción generada automáticamente

SUCURSALES

Solo definimos 10 sucursales para La Caja Arequipa.

Tabla

Descripción generada automáticamente

SOLICITUD PRESTAMO:

En esta parte las cuentas de los clientes quieren realizar solicitudes de prestamos. Por ejemplo, vemos que la cuenta de ID a 4 solicita un prestamo con un monto de 5000 y con la fecha

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Vemos un ejemplo de una modificacion de datos de la solicitud de ID 15, cambiamos el monto por 1000.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Aquí se muestra la funcionalidad de eliminar la solicitud de prestamo.

Interfaz de usuario gráfica, Aplicación, PowerPoint

Descripción generada automáticamente Tabla

Descripción generada automáticamente

EMPLEADOS:

En esta parte solo se mostrara los registrso de los empleados de la empresa

Interfaz de usuario gráfica

Descripción generada automáticamente

Asignamos un empleado.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Modificamos el dato por otro, cargo a cajero, y se cambiaa automaticamente.

Interfaz de usuario gráfica

Descripción generada automáticamente

Un ejemplo de eliminacion de un registro.

Interfaz de usuario gráfica

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

TRANSACCIONES:

Pasamos con las transacciones que son de retiro, deposito y transferencia de monto.

Retiro:

En esta parte vemos los registros de las transacciones hechas por nuestro grupo.

Interfaz de usuario gráfica

Descripción generada automáticamente

Mostramos un ejemplo de esta transaccion, insertamos los campos y colamos el monto de 120 para retirar.

Interfaz de usuario gráfica

Descripción generada automáticamente

Observamos que anteriormente teniamos nuestra lista de registros de las cuentas,recordemos que el cliente de ID 4 contaba con un saldo de 2000, aquí se muestra a continuacion:

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Cuando ya realizamos la transaccion de retiro de monto vemos que se descuenta su saldo del cliente.

Interfaz de usuario gráfica

Descripción generada automáticamente

Deposito:

Se mostrar a continuacion un ejemplo de registro de transaccion de Deposito en la cuenta de ID 3.

Interfaz de usuario gráfica

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Ahora vamos con los registros anteriores de los saldos de las cuentas. Ese cliente tenia un saldo de 1000.

Interfaz de usuario gráfica

Descripción generada automáticamente

Ahora cuando ya hicimos la transaccion de deposito, vemos que se a añadido dicho monto en su saldo.

Interfaz de usuario gráfica

Descripción generada automáticamente

Transferencia:

Ahora por ultimo vamos con la transacción de Transferencia de montos de una cuenta a otra.

Le damos click en el boton “Transferencia” y nos dirigira a otro formulario.

Interfaz de usuario gráfica

Descripción generada automáticamente

Vemos los registros de las cuentas y de las transacciones.

Primero se inserta los ID de la cuenta 1 que transferira un monto a la cuenta 2 y se selecciona una fecha para guardar la transaccion.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

En este ejemplo colocamos que la cuenta que transferira a la otra cuenta, son 3 y 11 en la cuenta 1 y 2 respectivamente.

Vemos que se inserta un monto de 50 con su respectiva fecha, luego finalmente se registra.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Se registro exitosamente.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Ahora vemos que se cambio los saldos por la transaccion.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Recomendaciones :

1. Optimización del Rendimiento:

Creo que sería beneficioso realizar una revisión exhaustiva del rendimiento de la aplicación. Podemos utilizar herramientas de perfilado de código para identificar áreas de mejora y optimizar las consultas a la base de datos. Esto mejoraría la velocidad de respuesta y la eficiencia general de la aplicación.

2. Seguridad de Datos

Dado el carácter sensible de la información financiera manejada por la Caja Arequipa, considero fundamental fortalecer las medidas de seguridad. Implementar encriptación de datos, validación de entrada rigurosa y una autenticación segura serían pasos esenciales para garantizar la integridad y confidencialidad de los datos.

3. Mejora de la Experiencia del Usuario (UX)

En mi opinión, sería valioso realizar mejoras en la experiencia del usuario. Podemos llevar a cabo pruebas de usabilidad, recopilar comentarios de los usuarios y realizar ajustes en la interfaz de usuario para asegurarnos de que la navegación sea intuitiva. Una interfaz más amigable contribuirá positivamente a la adopción y satisfacción del usuario.

Conclusiones :

1. Eficiencia a través de la Estructura en Capas:

Observo que la implementación de una arquitectura en capas ha resultado en una organización eficiente del código. La clara separación entre la capa de presentación, lógica de negocio y acceso a datos facilita el mantenimiento y la comprensión del sistema, permitiendo adaptaciones futuras de manera ágil.

2. Compromiso con la Seguridad y Fiabilidad:

Mi enfoque en la seguridad de datos y la implementación de prácticas robustas en todas las capas refleja mi compromiso con la integridad y confidencialidad de la información manejada. Esta atención es crucial, especialmente al trabajar con datos financieros como los de la Caja Arequipa.

3. Enfoque Holístico desde Desarrollo hasta Implementación:

Considero que la combinación de pruebas unitarias, documentación detallada y la consideración de prácticas de integración continua y despliegue continuo representa un enfoque holístico en el desarrollo de software. Este enfoque no solo asegura la calidad del código, sino que también facilita la colaboración y la entrega rápida y segura de nuevas versiones de la aplicación.

# XI. Anexos

Vertabelo: <https://my.vertabelo.com/public-model-view/HtEGSRDo4z0iM6JHtdGPYGliyYO5a0E3HKmOa67W9Qg8I3niSu6LPLQV9pZ1TWri?x=5020&y=5023&zoom=0.7456>

# XII. Bibliografía

Caja Municipal de Ahorro y Crédito de Arequipa (Caja Arequipa) (2022). *Memoria anual 2022*. <https://www.cajaarequipa.pe/memorias-institucionales/>