

# **Trabajo Práctico Integrador 2022 Banco BRULIM**

**Integrantes:** Bruno Jorge  
Limideiro Marcelo

**Docentes:** Lucas Salvatori / Mariano Martinez

**Breve descripción:**

El trabajo trata sobre el armado de un sistema informático de un sistema bancario.

El sistema bancario cuenta con una pantalla de login, una pantalla de bienvenida con un menú desplegable con las opciones del usuario.

Algunos de los movimientos que van a ser considerados son: transferencia entre cuentas, depósito en caja de ahorro/cuenta corriente, retiro de dinero, pago de impuestos, etc.

**Requerimientos funcionales:**

<b>Requerimiento N°: 1</b>	
<b>Fecha:</b>	<b>Analista Responsable:</b>
<b>Descripción;</b>	<b>Alta de usuario</b>
<b>Actores:</b>	<b>Administrador</b>
<b>Requerimiento dependiente:</b>	
<b>Secuencia:</b>	<b>El administrador crea un usuario al cliente a partir del apellido y sus nombres separados por un punto(.)</b>
<b>Consecuencia de rechazo 1:</b>	
<b>Consecuencia de rechazo 2:</b>	

<b>Requerimiento N°:2</b>	
<b>Fecha:</b>	<b>Analista Responsable:</b>
<b>Descripción;</b>	<b>Login usuario</b>
<b>Actores:</b>	<b>Administrador, cliente</b>
<b>Requerimiento dependiente: requerimiento n°1</b>	
<b>Secuencia:</b>	<p><i>El usuario ingresa con un nombre de usuario y una contraseña para que el sistema valide los datos.</i></p> <p><i>La contraseña debe ser alfanumérica con un mínimo de 8 caracteres y un máximo de 12.</i></p> <p><i>Si el usuario y la contraseña son correctas el login se produce.</i></p>
<b>Consecuencia de rechazo 1:</b>	<i>El usuario ingresa de forma errónea la contraseña. La pantalla muestra un mensaje de “contraseña incorrecta”. Retorna la pantalla para el ingreso del usuario y contraseña.</i>
<b>Consecuencia de rechazo 2:</b>	<i>El usuario ingresa de forma errónea el usuario. La pantalla muestra un mensaje de “usuario incorrecto”. Retorna la pantalla para el ingreso del usuario y contraseña.</i>

<b>Requerimiento N°: 3</b>	
<b>Fecha:</b>	<b>Analista Responsable:</b>
<b>Descripción;</b>	<b>El cliente realiza la apertura de una cuenta</b>
<b>Actores:</b>	<b>Cliente</b>
<b>Requerimiento dependiente:, requerimiento n°2</b>	
<b>Secuencia:</b>	<b>Ingresa a la opción de apertura de cuenta. Selecciona el tipo de cuenta a abrir Caja de Ahorro o Cuenta Corriente. El sistema devuelve por pantalla la confirmación de alta de cuenta con su respectivo número.</b>
<b>Consecuencia de rechazo 1:</b>	
<b>Consecuencia de rechazo 2:</b>	

<b>Requerimiento N°: 4</b>	
<b>Fecha:</b>	<b>Analista Responsable:</b>
<b>Descripción;</b>	<b>Cierre de cuenta</b>

<b>Actores:</b>	<b>Cliente</b>
<b>Requerimiento dependiente: requerimiento n°3</b>	
<b>Secuencia:</b>	<i>Ingresa a la opción de cierre de cuenta. Selecciona la cuenta a cerrar por su número. Por pantalla el sistema pide confirmación para eliminar la cuenta. Se valida que la cuenta esté en 0 (cero) para poder realizar el cierre.</i>
<b>Consecuencia de rechazo 1:</b>	<i>La cuenta tiene saldo. El sistema muestra por pantalla una advertencia que no puede realizar la baja porque la cuenta tiene saldo.</i>
<b>Consecuencia de rechazo 2:</b>	

### **Requerimientos no funcionales:**

- La persistencia en la base de datos necesita una conexión local suministrada por el servicio MySQL.
- El sistema de versionado estará a cargo del servicio online GitHub

### **Diagrama de flujo de datos:**

<https://www.figma.com/file/DG63hUDkQIAAn1HhLfZtYDv/TP-FINAL-LUCAS?node-id=0%3A1>

### ***Diccionario de Datos:***

LOGIN = {@usuario + @contraseña}

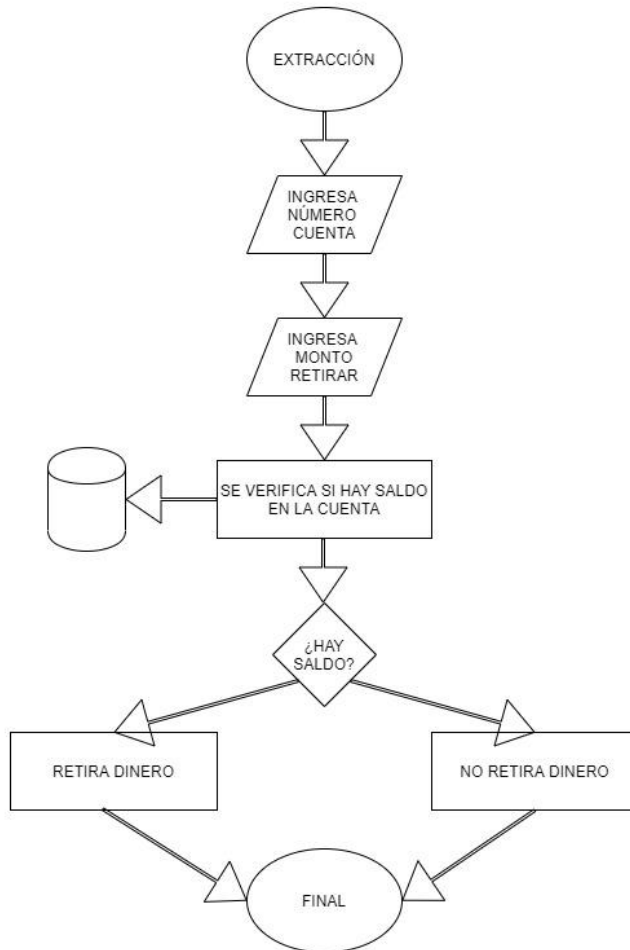
CARGA INDIVIDUO = @CUIL/CUIT + (DNI) + (Nombre) + (Apellido) + @Teléfono +  
@Email + @Nombre-de-Usuario + @Password

CARGA PyME = @CUIT + (RazónSocial) + @Teléfono + @Email + @Nombre-de-Usuario  
+ @Password

CREAR CUENTA = [ Caja\_Ahorro\_Común | Caja\_Ahorro\_Saldo\_Retenido |  
Cuenta\_Corriente\_Pesos | Cuenta\_Corriente\_Dólares ]

### **Especificación de Procesos (explotados en DFD):**

Este proceso muestra como es el procedimiento para realizar una extracción de dinero.



Este proceso muestra como es el procedimiento para realizar una transferencia de dinero de una cuenta propia a una cuenta extranjera.

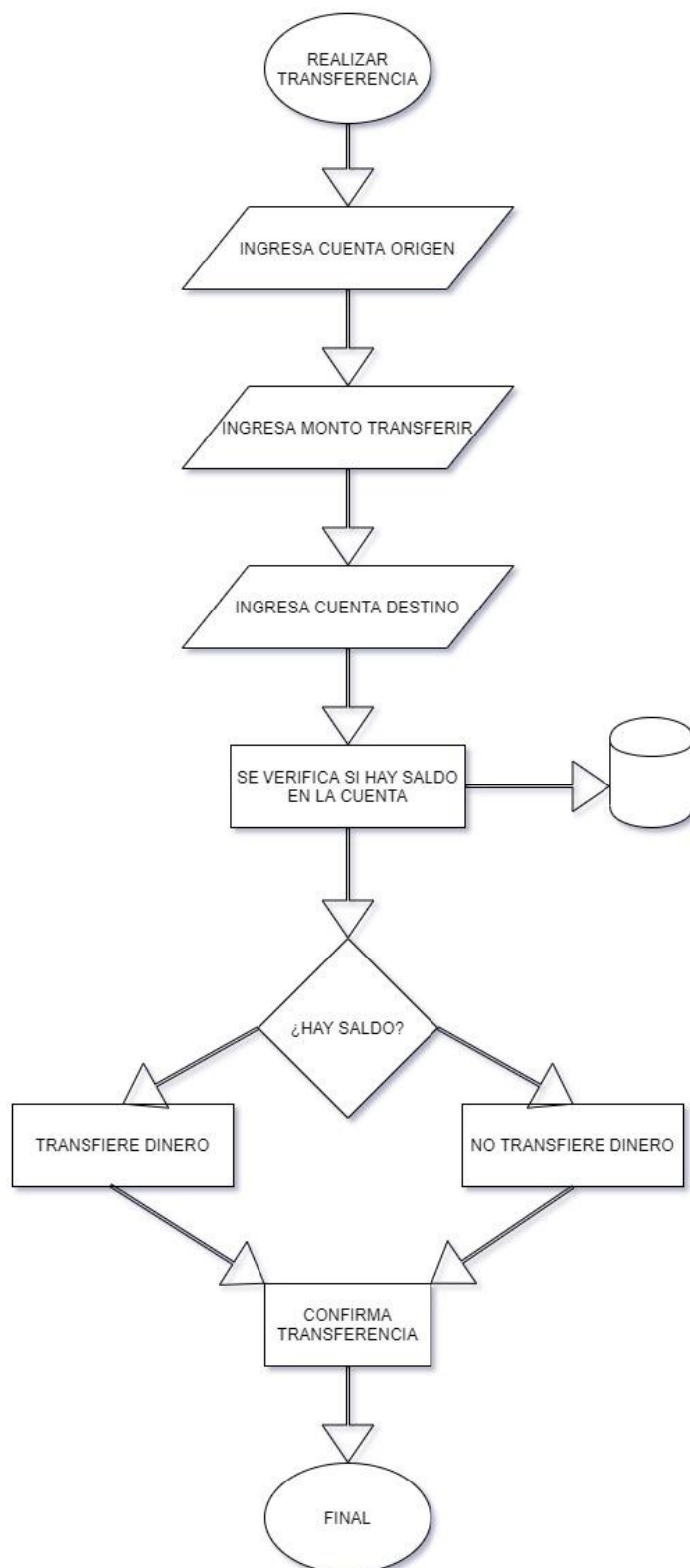
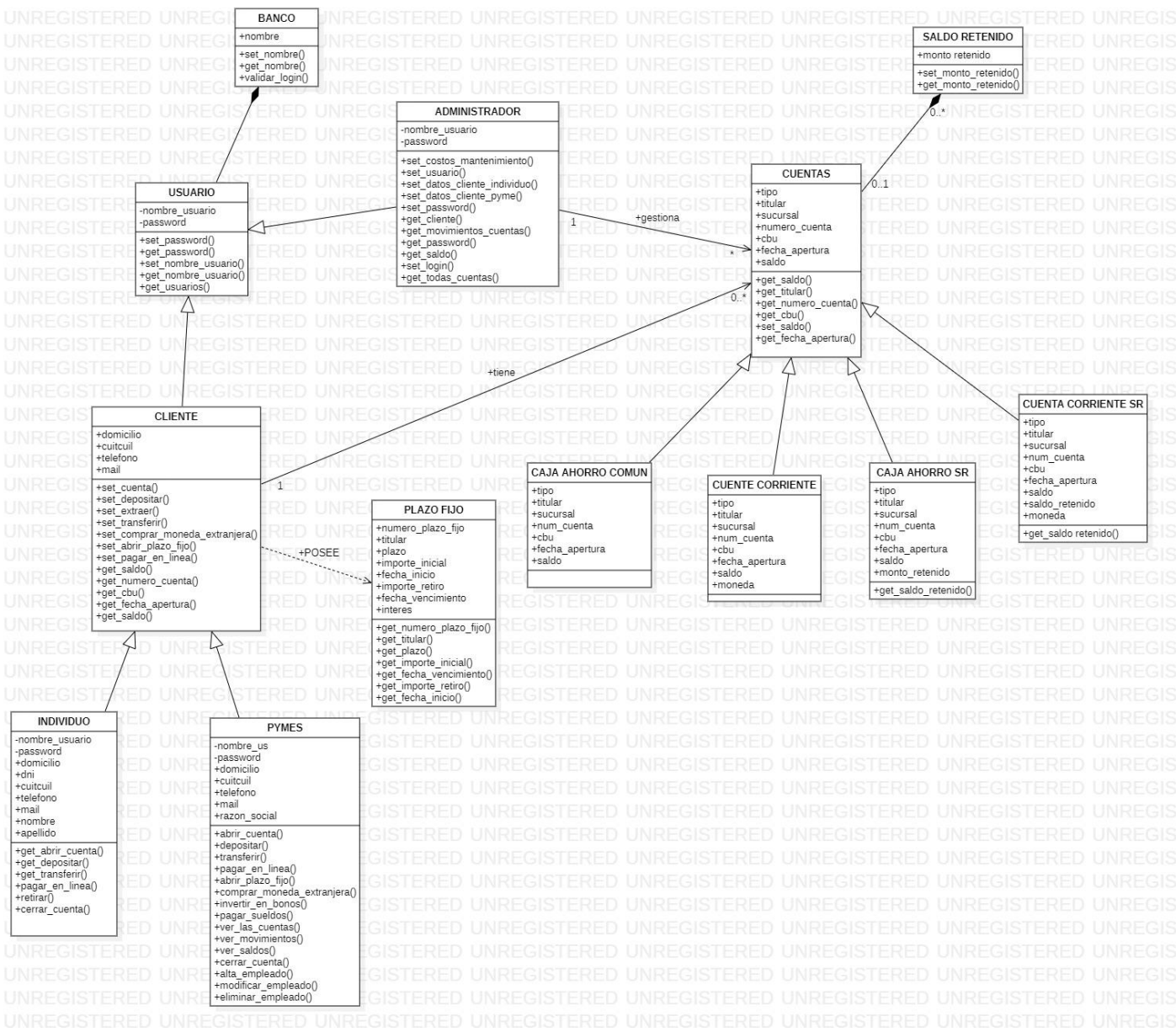
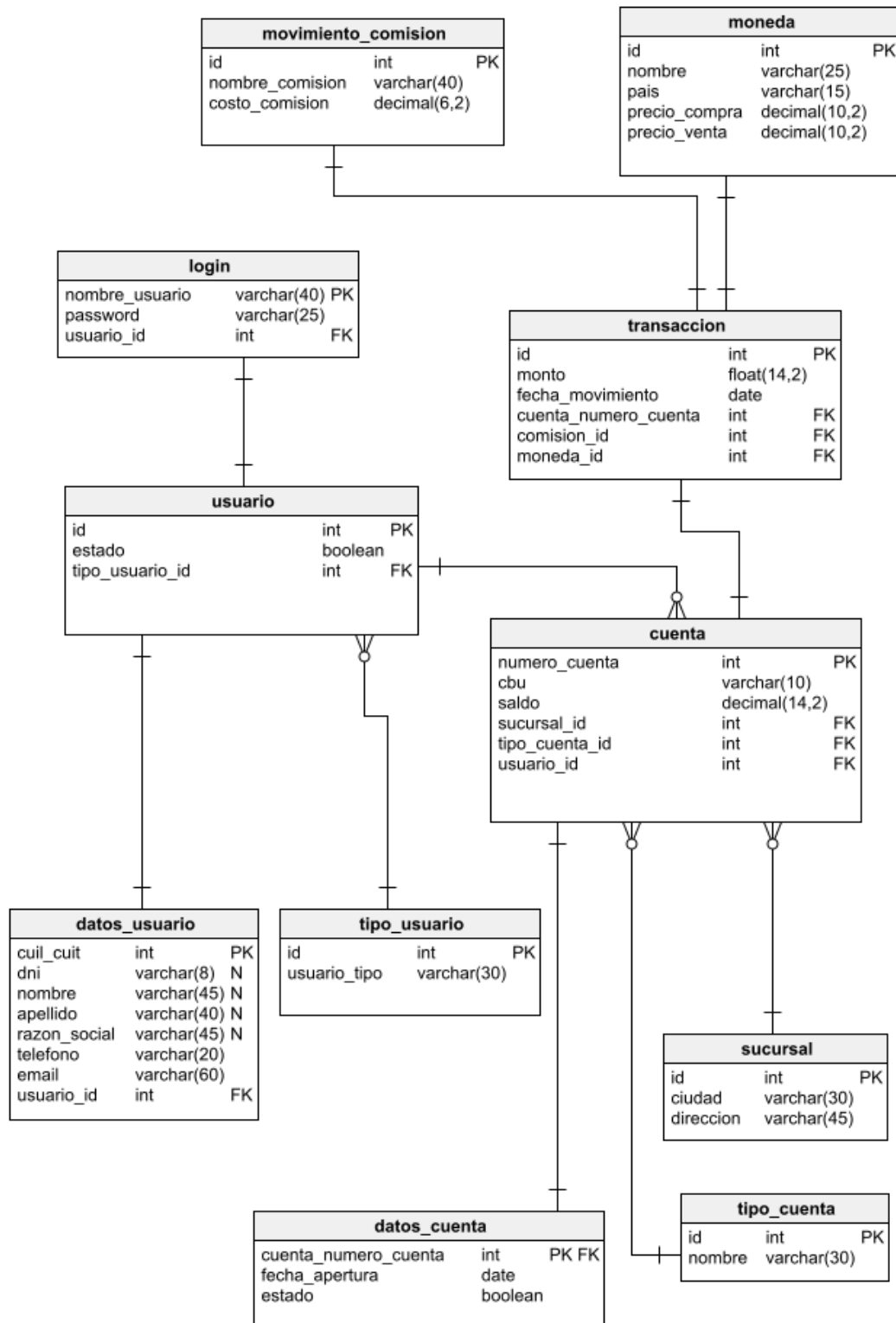




Diagrama de Clases:



## Diagrama de Entidad Relación:



Nuestro DER se encuentra normalizado hasta la 3FN.

No presenta redundancia de datos en ninguna de sus tablas. La dependencia de los datos se condicen con cada clave primaria de cada tabla.

Cabe aclarar que decidimos separarla tabla USUARIO de la de DATOS\_USUARIO justamente para aplicar la 3FN, el estado del usuario y el tipo\_usuario (que viene de la tabla TIPO\_USUARIO) si dependen del id de la tabla, pero los datos del usuario pueden no depender de dicho id(PK).

Una vez separada la tabla los datos del usuario dependen de la PK en este caso CUIL/CUIT.

### ***Análisis de patrones arquitectónicos:***

Para nuestro sistema elegimos la arquitectura de software llamada Modelo-vista-controlador (MVC).

MVC propone la construcción de tres componentes distintos, el modelo, la vista y el controlador.

El modelo define qué datos debe contener la aplicación. Si el estado de estos datos cambia, el modelo generalmente notificará a la vista (para que la pantalla pueda cambiar según sea necesario).

La vista define cómo se deben mostrar los datos de la aplicación. En nuestra aplicación bancaria, la vista definirá cómo se presenta la lista al usuario y recibirá los datos para mostrar desde el modelo.

El controlador contiene una logica que actualiza el modelo y/o vista en respuesta a las entradas de los usuarios de la aplicación.

Entonces, por ejemplo, nuestra aplicación contiene formularios de entrada y botones que nos permiten crear y/o eliminar cuentas, listar movimientos y saldos. Estas acciones requieren que se actualice el modelo, por que la entrada se envía al controlador, que luego manipula el modelo según corresponda, que luego envía datos actualizados a la vista.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

**Modelo de Vistas:**

A Web Page

https://

BANCO BRULIM

...

Operaciones

Cargar Individuo

Cargar Pyme

Listar Saldos

Listar Cuentas

Listar Usuarios

Listar Movimientos

Costos / Comisiones

SALIR

Listado de Todas las Cuentas

id Cliente	Nombre	Apellido	Razón Social	Nº Cuenta	Saldo
1	Marcelo	Limideiro	None	Caja Ahorro	500.00
1	Marcelo	Limideiro	None	Cuenta Corriente	6500.00
1	Marcelo	Limideiro	None	Caja Ahorro	0.00
1	Marcelo	Limideiro	None	Caja Ahorro	0.00
1	Marcelo	Limideiro	None	Caja Ahorro	0.00
1	Marcelo	Limideiro	None	Cuenta Corriente	0.00
1	Marcelo	Limideiro	None	Caja Ahorro	0.00
1	Marcelo	Limideiro	None	Caja Ahorro	0.00
1	Marcelo	Limideiro	None	Caja Ahorro	0.00
1	Marcelo	Limideiro	None	Caja Ahorro	0.00

A Web Page

X

https://

Q

BANCO BRULIM

...

Operaciones

Cargar Individuo

Cargar Pyme

Listar Saldos

Listar Cuentas

Listar Usuarios

Listar Movimientos

Costos / Comisiones

SALIR

Listado de Todas los Saldos

id Cliente	Nombre	Apellido	Razón Social	Nº Cuenta	Saldo
1	Marcelo	Limideiro	None	1	500.00
1	Marcelo	Limideiro	None	13	0.00
1	Marcelo	Limideiro	None	12	0.00
1	Marcelo	Limideiro	None	11	0.00
1	Marcelo	Limideiro	None	10	0.00

A Web Page

X

https://

Q

BANCO BRULIM

...

Operaciones

Cargar Individuo

Cargar Pyme

Listar Saldos

Listar Cuentas

Listar Usuarios

Listar Movimientos

Costos / Comisiones

SALIR

Carga de Cliente PyME

Ingrese los Datos

CUIT

Razón Social

Teléfono

Email

Nombre de Usuario

Password

Enviar

A Web Page

X

https://

Q

BANCO BRULIM ...

OperacionesCrear CuentaListar CuentasListar SaldosListar MovimientosREALIZAR DepósitoRealizar TransferenciaRealizar RetiroSALIR

Crear una Cuenta

Escoja un tipo de cuenta

Caja Ahorro Común

Caja Ahorro Saldo Retenido

Cuenta Corriente Pesos

Cuenta Corriente Saldo Retenido

Cuenta Corriente Dólares

A Web Page

X

https://

Q

BANCO BRULIM ...

OperacionesCrear CuentaListar CuentasListar SaldosListar MovimientosREALIZAR DepósitoRealizar TransferenciaRealizar RetiroSALIR

Listado de Movimientos

Ingrese el Número de Cuenta

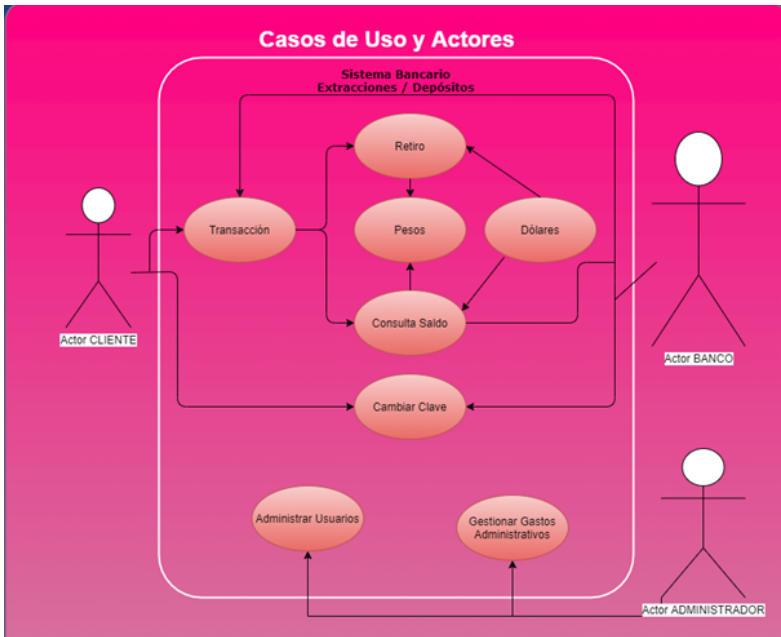
1

Aceptar

Id Movimiento	Monto	Fecha Movimiento	Tipo Comisión	Tipo Moneda	Costo Comisión
1	420.0	2022-01-01	Transferencia Caja Ahorro	Peso	60.00
2	310.0	2022-01-11	Transferencia Caja Ahorro	Peso	60.00

Casos de Uso:

Caso de Uso 1:



Caso de Uso 2:



<b>NOMBRE:</b>	<b>LOGIN</b>
<b>DESCRIPCION:</b>	El usuario registrado ingresa su usuario y contraseña
<b>ACTORES:</b>	Administrador / clientes
<b>PRECONDICIONES:</b>	El usuario debe estar cargado en la bbdd
<b>FLUJO NORMAL:</b>	El usuario ingresa a la App
<b>FLUJO ALTERNATIVO:</b>	El usuario y/o la contraseña son incorrectas y regresa a la página login
<b>POSCONDICIONES:</b>	El sistema lo redirecciona a la pagina para realizar operaciones
<b>NOMBRE:</b>	<b>ALTA DE CLIENTE INDIVIDUO</b>
<b>DESCRIPCION:</b>	El administrador carga un nuevo cliente individuo
<b>ACTORES:</b>	Administrador
<b>PRECONDICIONES:</b>	Administrador logeado correctamente
<b>FLUJO NORMAL:</b>	El nuevo cliente es cargado en la bbdd
<b>FLUJO ALTERNATIVO:</b>	El cliente ya existe- No permite la carga
<b>POSCONDICIONES:</b>	El cliente queda habilitado para operar en el banco
<b>NOMBRE:</b>	<b>ALTA DE CLIENTE PYME</b>
<b>DESCRIPCION:</b>	El administrador carga un nuevo cliente pyme
<b>ACTORES:</b>	Administrador
<b>PRECONDICIONES:</b>	Administrador logeado correctamente
<b>FLUJO NORMAL:</b>	El nuevo cliente es cargado en la bbdd
<b>FLUJO ALTERNATIVO:</b>	El cliente ya existe- No permite la carga
<b>POSCONDICIONES:</b>	El cliente queda habilitado para operar en el banco
<b>NOMBRE:</b>	<b>CREAR CUENTA</b>
<b>DESCRIPCION:</b>	El cliente puede crear una nueva cuenta en el banco
<b>ACTORES:</b>	Cliente individuo- cliente pyme
<b>PRECONDICIONES:</b>	Cliente logeado correctamente
<b>FLUJO NORMAL:</b>	El cliente elije qué tipo de cuenta desea abrir- La cuenta es cargada en la bbdd
<b>FLUJO ALTERNATIVO:</b>	El cliente no deposita el SALDO RETENIDO-Debe abrir otro tipo de cuenta
<b>POSCONDICIONES:</b>	La cuenta queda habilitada para operar



<b>NOMBRE:</b>	<b>REALIZAR UN DEPÓSITO</b>
<b>DESCRIPCION:</b>	El cliente realiza un depósito en su cuenta
<b>ACTORES:</b>	Cliente individuo / cliente pyme
<b>PRECONDICIONES:</b>	Cliente logeado correctamente - Poseer una cuenta propia
<b>FLUJO NORMAL:</b>	Ingresa el monto y se acredita en la cuenta indicada.
<b>FLUJO ALTERNATIVO:</b>	No ingresa el monto, sale de la opcion depositar
<b>POSCONDICIONES:</b>	El saldo de esa cuenta es actualizado- Envía al usuario al menú principal.

<b>NOMBRE:</b>	<b>REALIZAR UNA EXTRACCIÓN</b>
<b>DESCRIPCION:</b>	El usuario extrae dinero de su cuenta
<b>ACTORES:</b>	Cliente individuo / cliente pyme
<b>PRECONDICIONES:</b>	Cliente logeado correctamente - Haber realizado un depósito
<b>FLUJO NORMAL:</b>	El cliente ingresa el monto a retirar y el n° de cuenta
<b>FLUJO ALTERNATIVO:</b>	La cuenta ingresada no es de su propiedad- El usuario individuo no dispone del saldo suficiente
<b>POSCONDICIONES:</b>	El saldo es actualizado - Envía al usuario al menú principal

<b>NOMBRE:</b>	<b>REALIZAR UNA TRANSFERENCIA</b>
<b>DESCRIPCION:</b>	El usuario transfiere dinero de una cuenta de su propiedad hacia cualquier otra cuenta
<b>ACTORES:</b>	Cliente individu / Cliente pyme
<b>PRECONDICIONES:</b>	Cliente logeado correctamente - Poseer una cuenta propia- Haber realizado un depósito
<b>FLUJO NORMAL:</b>	Ingresa el n°de cuenta origen, ingresa el importe a transferir, ingresa el ° de cuenta destino
<b>FLUJO ALTERNATIVO:</b>	El usuario no es el titular de la cuenta origen ingresada. La cuenta origen no existe. -La cuenta destino no existe. -La cuenta origen no dispone de saldo suficiente
<b>POSCONDICIONES:</b>	EL saldo de la cuenta origen y de la cuenta destino son actualizados

<b>NOMBRE:</b>	<b>VER SALDO DE LA CUENTA</b>
<b>DESCRIPCION:</b>	El usuario ve el saldo actual de todas o alguna de sus cuentas.
<b>ACTORES:</b>	Cliente individu / Cliente pyme
<b>PRECONDICIONES:</b>	Cliente logeado correctamente - Poseer una cuenta propia
<b>FLUJO NORMAL:</b>	El usuario ingresa el n° de cuenta o elije la opción de todas las cuentas
<b>FLUJO ALTERNATIVO:</b>	No es titular de la cuenta ingresada.- No posee cuentas
<b>POSCONDICIONES:</b>	El sistema muestra el/los saldos solicitados y regresa al menú principal.

<b>NOMBRE:</b>	<b>VER LOS MOVIMIENTOS</b>
<b>DESCRIPCION:</b>	El usuario visualiza el historial de movimientos de una cuenta.
<b>ACTORES:</b>	Cliente individu / Cliente pyme
<b>PRECONDICIONES:</b>	Cliente logeado correctamente - Poseer una cuenta propia
<b>FLUJO NORMAL:</b>	El usuario ingresa un n° de cuenta
<b>FLUJO ALTERNATIVO:</b>	No es titular de la cuenta ingresada.- No posee cuentas
<b>POSCONDICIONES:</b>	El sistema muestra el/los movimientos de la cuenta solicitada y regresa al menú principal.

<b>NOMBRE:</b>	<b>LISTAR SALDOS</b>
<b>DESCRIPCION:</b>	El administrador lista los saldos de todas las cuentas del banco
<b>ACTORES:</b>	Administrador
<b>PRECONDICIONES:</b>	Administrador logeado correctamente
<b>FLUJO NORMAL:</b>	El administrador elije la opción y el sistema muestra una lista con todos los saldo
<b>FLUJO ALTERNATIVO:</b>	El administrador no se logeo de manera adecuada
<b>POSCONDICIONES:</b>	El administrador ve todos los saldos y vuelve al menú principal

<b>NOMBRE:</b>	<b>SALIR</b>
<b>DESCRIPCION:</b>	El usuario se desloga y sale del sistema
<b>ACTORES:</b>	Administrador / clientes
<b>PRECONDICIONES:</b>	Haberse logeado correctamente
<b>FLUJO NORMAL:</b>	El usuario elije la opción salir.
<b>FLUJO ALTERNATIVO:</b>	
<b>POSCONDICIONES:</b>	El sistema lo renderiza a la página del login

### ***Especificación de lenguajes y tecnologías:***

El lenguaje utilizado en el proyecto es Python junto al framework Flask. Flask está escrito en python y permite crear aplicaciones web. Está basado en la especificación WSGI y el motor de templates Jinja2.

Para la base de datos se utilizó un sistema de gestión de bases relacional MySQL.

En nuestro caso, utilizamos el paquete de herramientas XAMMP para la conexión con la base de datos (puerto 3306).

### ***Notas y Aclaraciones:***

Las funcionalidades habilitadas son:

- Login con chequeo en base de datos si el usuario y/o la password son validas
- Distinción en el login entre un usuario Administrador y un usuario PyME o Individuo
- Dependiendo del tipo de usuario se carga el menú correspondiente
- Administrador:
  - Carga de Individuo (insert en base datos)
  - Carga de PyME (insert en base datos)
  - Listar Saldos (select en base datos)
  - Listar Cuentas (select en base datos)
  - Listar Usuarios (select en base datos)
- Individuo:
  - Crear Caja Ahorro (la cuenta se crea, pero falta coordinar el insert en la tabla datos\_cuenta para que al realizar la busqueda, traiga los datos cargados)
  - Crear Cuenta Corriente Pesos
  - Listar Cuentas (falta corregir el select de cuentas creadas desde la app)
  - Listar Saldos
  - Listar Movimientos
- La opción SALIR del menú lleva a la pagina de login
- Sistema implementado que evita que un usuario Individuo, copie el link del Administrador y realice operaciones no permitidas

- Se modularizó el controlador app.py y se separaron las rutas de los usuarios con respecto a las del Administrador.
- Carpeta Views contiene las vistas
- Carpeta Models contiene los modelos
- Test de funcionalidad no se logró terminar
- Git se utilizó para el versionado del proyecto y el trabajo en equipo