

# **Memoria Escrita Práctica 1**

Programación Orientada a Objetos

**Proyecto:** Interfaz bancaria **Wunallet**

Grupo 1 Equipo 7

César Augusto Ospina Muñoz

Diego Ospina Ramírez

David Cardona Duque

Daniel Escobar David

# Índice

1. Descripción general de la solución (análisis, diseño e implementación)
2. Diagrama UML
3. Descripción de donde se implementan las características de POO (los detalles se pueden ver en el numeral 6). Se debe indicar dónde y cómo se implementaron
4. Descripción de cada una de las 5 funcionalidades donde se enuncie su funcionamiento, que objetos intervienen en su implementación con un breve modelo de la secuencia del proceso, y UNA captura de pantalla con los resultados que presenta al usuario
5. Manual de usuario donde se indique como testear cada funcionalidad junto a los datos.

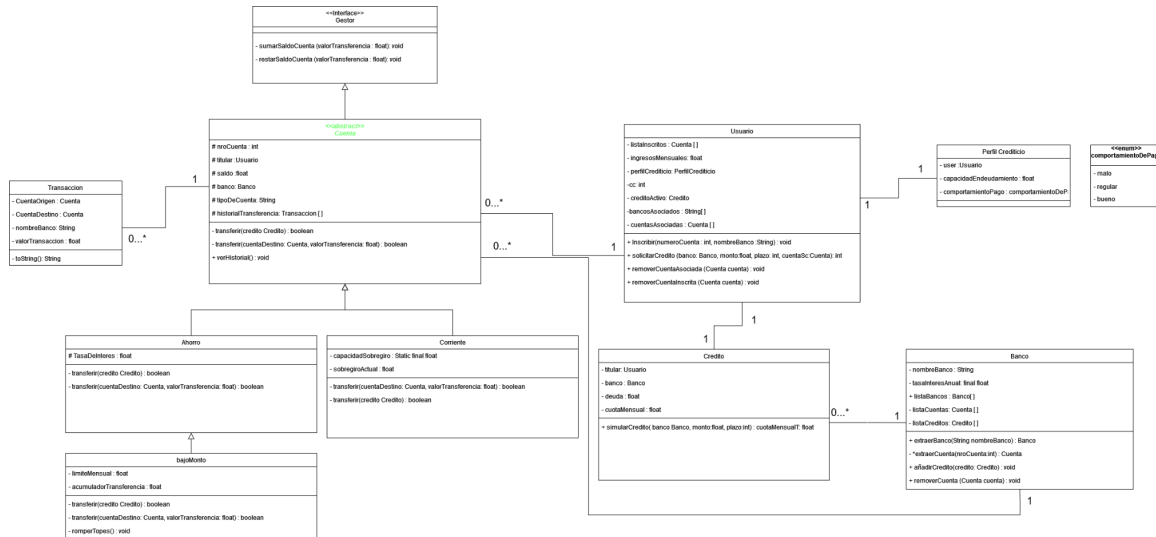
1. Wunallet es un software interbancario inspirado en las aplicaciones bancarias convencionales, como por ejemplo Nequi y ahorro a la mano, pero que a diferencia de estas integra múltiples bancos en su portafolio; lo que permite, entre otras cosas, hacer transferencias entre bancos o solicitudes de crédito a diferentes entidades, desde una misma interfaz.

Esto está logrado a nivel de diseño al describir todo el entorno interbancario mediante 10 clases y una interfaz de gestión. Estas clases, en detalle, son:

- Banco: Cada instancia de esta clase representa uno de los bancos integrados al software, y en sus atributos almacenan toda la información sobre sus productos e identidad.
- Usuario: Como piedra angular del diseño, cada instancia de esta clase representa un usuario. Es su información y operaciones las que dan sentido y cohesión a todas las demás clases, y es el único parámetro que todas las funcionalidades deben garantizar.
- Credito: Son los objetos creados cuando a un usuario se le aprueba su solicitud de crédito. Su función va desde almacenar los detalles del préstamo y a qué banco pertenece, hasta ofrecer simulaciones de crédito.
- Banquero: De forma análoga al caso real, esta clase gestiona la interfaz gráfica, los inputs del usuario, la ejecución de funcionalidades, y los retornos en pantalla.
- PerfilCrediticio: Cada instancia está asociada a un usuario, y describe precisamente el perfil crediticio de éste. Es usada por los bancos para determinar si una solicitud de crédito se aprueba o rechaza.
- Transaccion: Es una clase que solamente tiene atributos de tal forma que, en conjunto, permiten reconstruir a detalle la información sobre una operación bancaria.
- Cuenta: Como clase abstracta se encarga de dar el cuerpo, a nivel de atributos y métodos, a sus clases hijas: Ahorro, Corriente y BajoMonto. Siempre están asociadas a un banco y usuario.
- Corriente: Siendo una clase hija de Cuenta, se diferencia por incluir la capacidad de sobregirarse, de tal forma que el usuario puede realizar transferencias de mayor cantidad que las de su saldo.
- Ahorro: Es el producto financiero más simple, diferenciándose de su clase padre Cuenta al ofrecer una tasa de interés que producirá ingresos en el tiempo.
- BajoMonto: Es el resultado de especializar las cuentas Ahorro, ya que si bien disponen de los mismos atributos, sus métodos están limitados a un tope máximo mensual que se puede transferir.

La implementación de la aplicación se hizo mediante una interfaz simple de consola, donde el usuario puede seleccionar una de las 5 funcionalidades disponibles y dentro de estas elegir entre más opciones que complementan o especifican su funcionamiento, de esta manera el usuario podrá acceder una amplia variedad de operaciones interbancarias desde una misma interfaz gráfica.

## 2. Imagen del diagrama UML



## 3.

1. La clase abstracta del diseño corresponde a la clase **Cuenta**, que a su vez contiene dos métodos abstractos y sobrecargados:

**abstract transferir(Cuenta cuentaDestino, float ValorTransferencia)**

**abstract transferir(Credito credito)**

Estos métodos ejecutan las comprobaciones y actualizaciones de saldo correspondientes al realizar una transferencia. Este proceso depende del tipo de cuenta que lo ejecute, y por eso debe ser un método abstracto.

2. La interfaz **Gestor** es la encargada de solicitar que se declaren métodos para, precisamente, gestionar los productos bancarios. Es implementada en la clase **Cuenta** ya que sus clases hijas requieren hacer operaciones de ajuste de saldo.
3. Las clases **Corriente** y **Ahorro** heredan de la clase abstracta **Cuenta**, y a su vez **BajoMonto** hereda de **Ahorro**, ya que toda cuenta de BajoMonto es esencialmente una cuenta de Ahorro con limitaciones en sus métodos y atributos.
4. Los métodos sobrecargados **transferir**, heredados desde la clase **Cuenta**, aplican ligadura dinámica al estar definida su invocación en base a si la instancia es de **Ahorro** o de **BajoMonto**.

**transferir(Cuenta cuentaDestino, float ValorTransferencia)**

**transferir(Credito credito)**

Por las limitaciones impuestas a las cuentas de **BajoMonto**, los métodos deben hacer verificaciones diferentes para realizar las transferencias de pago de crédito y entre cuentas.

5. En la clase **Banco**, el atributo **listaBancos** es un método estático usado para almacenar los bancos que han sido creados y hacen parte del 'portafolio' de la aplicación.  
Adicionalmente, el método **extraerBanco(String nombreBanco)** también de la clase **Banco**, es estático ya que se usa para obtener del atributo anterior el objeto banco a partir de su nombre.
6. La constante *costoRomperTopes* implementada en la clase **BajoMonto** es la que permite verificar si se puede ejecutar o no la funcionalidad **Romper topes** en base al saldo de la cuenta.
7. **Protected:** Es usado en la clase **Cuenta** y **Ahorro** con todos sus atributos, ya que ambas clases heredan y se quería dar accesibilidad a sus clases hijas.

**Private:** Es la visibilidad predilecta de los atributos de instancia, y se aplicó en (algunos o todos) los atributos de las clases: **BajoMonto**, **Corriente**, **Crédito**, **Banco**, **PerfilCrediticio**, **Transaccion** y **Usuario**.

**Public:** En los atributos está presente en la clase **Banco** ya que por diseño el atributo *listaBancos* debe ser público y estático para determinar el portafolio de la aplicación. Sin embargo en cuanto a los métodos su visibilidad predilecta fue *public* y todos los métodos, en todas las clases, lo implementaron de esta manera.

8.
  - 8.1. Sobrecarga del constructor: La primera sobrecarga del constructor se implementó en el módulo gestorAplicacion dentro del paquete infoClientes, en la clase transaccion, esto debido a que dentro de la lógica de la aplicación se tienen 2 tipos de transacciones posibles, la primera la transacción es de una cuenta bancaria a otra, la cual implica parámetros como la cuenta de destino, y el segundo tipo de transacción es la que se encarga de pagar créditos asociados al usuario, la cual implica parámetros como el banco con el cual se tiene el crédito.

La segunda sobrecarga de constructor se implementó en el módulo gestorAplicacion dentro del paquete productosFinancieros, en la clase Ahorro, esto debido a que una cuenta de ahorro se crea con historial vacío y cuando se rompen topes se debe heredar el historial de la cuenta pasada.

Sobrecarga de métodos: La sobrecarga de métodos se implementa en el módulo gestorAplicacion, dentro del paquete productosFinancieros en todas las clases que heredan de Cuenta, esta sobrecarga se da mediante el método transferir debido a

que dentro de la aplicación existen 2 tipos de transferencias, una que va dirigida a otra cuenta bancaria y por lo tanto necesita parámetros como la cuenta de destino y el otro tipo de transferencia que está enfocado a pagar créditos asociados con el usuario y necesita el parámetro crédito.

8.2. this de desambiguación: Los this que desambiguan se utilizaron en la gran mayoría de constructores de las clases con el objetivo de diferenciar el parámetro pasado al constructor con el atributo de la clase, ahora bien, para cumplir con este requisito se mencionan 2 casos específicos, estos se implementaron en el módulo gestorAplicacion, dentro del paquete productosFinancieros en el constructor de la clase bajoMonto, se desambigua el atributo limiteMensual con el parámetro limiteMensual y el atributo acumuladorTransferencia con el parametro acumuladorTransferencia.

this(): El uso de este this se implementó en el módulo gestorAplicacion, dentro del paquete productosFinancieros, en la clase ahorro en la sobrecarga de métodos, este this se utiliza para setear como un historial vacío con el otro constructor y crear una cuenta de ahorro desde 0.

8.3. Caso de enumeración: El caso de numeración fue implementado en el módulo gestorAplicacion dentro del paquete infoClientes, en la clase de enumeración comportamientoDePago, este enum consiste de los niveles de comportamiento de pago de cada usuario, los cuales están ranqueados desde bueno a malo con números del 1 al 3, se utilizó para tener un nivel de referencia a la hora de solicitar un crédito a una entidad bancaria. Se tiene dentro de esta clase el método comportamientoDePago el cual retorna de manera aleatoria una de las categorías anteriores para así tomar decisiones a la hora de aprobar un crédito.

4.

## Inscribir cuenta

### *Funcionamiento:*

Esta funcionalidad le permite a un usuario inscribir una cuenta a la que transfiere frecuentemente para evitar ingresar los datos en cada ocasión, y, a su vez, aumentar la cantidad de dinero que puede enviar en cada transacción hacia dicha cuenta.

Está relacionada con las clases **Usuario**, **Cuenta** y sus clases hijas; y **Banco**.

### *Secuencia:*

Una vez seleccionada se le pide al usuario que seleccione e ingrese de qué banco es la cuenta a inscribir, el tipo de cuenta y el número de cuenta así como la cédula del titular de dicha cuenta. En base a esta información se verifica en el **banco** seleccionado que exista

una **cuenta** con todos los datos ingresados. De ser así la cuenta es extraída y vinculada al **usuario**.

*Captura de pantalla:*

```
-----
#   Inscribir cuenta
-----
- Selecccion del banco
-----
1. Unalombia
2. PooBanco
3. QuitaVivienda
-----
- Seleccione de que banco es la cuenta que desea inscribir: 1
-----
R / Usted selecciono Unalombia
-----
- Selección tipo de cuenta
-----
1. Cuenta de bajo monto
2. Cuenta de ahorros
3. Cuenta corriente
4. Volver al menú de funcionalidades
Seleccione el tipo de cuenta: 1
-----
R / Usted seleccionó bajoMonto
-----
- Inserción del número de cuenta
-----
Ingrese el número de la cuenta: 26
-----
- Inserción del número de cédula
Ingrese el número de cedula: 98
-----
-----
R / Las cuentas inscritas del usuario 10 son: 26
-----
```

## Romper topes

*Funcionamiento:*

Esta funcionalidad es exclusiva de usuarios que cuentan con por lo menos una cuenta de BajoMonto. Su funcionalidad es romper las limitaciones en la cantidad de dinero que se puede usar cada mes con las cuentas de tipo BajoMonto al reemplazar dicha cuenta por una de Ahorro, migrando toda la información y saldos en el proceso.

Está relacionada con las clases **Usuario**, **Cuenta**, **BajoMonto**, **Ahorro**, **Gestor (interface)** y **Banco**.

*Secuencia:*

Una vez escogida la funcionalidad desde la interfaz se realiza una comprobación de que el **usuario** tenga al menos una cuenta de **bajo monto**. De ser así se le indica al usuario en qué consiste el proceso que va a realizar y se le pide que seleccione la **cuenta** objetivo. Tras verificar que la cuenta cumpla las condiciones (con ayuda de la **interface**) para realizar la actualización, se procede a crear una cuenta de **ahorro** que se inicializa con los datos de la cuenta anterior y finalmente se realizan las limpiezas y desasignaciones pertinentes de la cuenta anterior desde el **banco**.

*Captura de pantalla:*

```
# Romper topes
-----
¡Recuerde! El procedimiento de romper topes consiste en transformar su cuenta de tipo Bajo monto,
a una cuenta de ahorros convencional, eliminando las limitaciones de este tipo de cuentas.
Este proceso tiene un costo de 15.000 pesos que pagará una única vez.
-----
- Selección de cuenta
-----
1 Cuenta 69
2 Cuenta 26
-----
- Para continuar seleccione la cuenta de bajo monto que desea transformar: 1
R / Usted seleccionó: 69
-----
R / Tu solicitud ha sido aprobada, se descontarán $15.000 de tu saldo para realizar el proceso. Espera un momento...
-----
R / Tu cuenta ha sido actualizada y ahora no tiene topes.
R / Tu nueva cuenta de ahorros ahora tiene un saldo de: 985000.0.
-----
```

## Solicitar crédito

*Funcionamiento:*

Permite que un usuario solicite un crédito con cualquier banco existente, brindando la oportunidad de escoger el monto y plazo del crédito. La solicitud puede ser aprobada o rechazada, y en caso de aprobación será depositado a la cuenta que el usuario indicó.

La funcionalidad está relacionada con la clase **Usuario**, **Banco**, **Cuenta** y sus hijas, **Credito** y **PerfilCrediticio**.

*Secuencia:*

Una vez seleccionada la funcionalidad desde la interfaz se realiza una comprobación de que el **usuario** no tenga ningún otro crédito activo. De ser así, procede a seleccionar el **banco** con el que desea adquirir el crédito, escoge la cuenta a la que será depositado en caso de ser aprobado, y por último ingresa el monto y plazo de interés. Con estos datos se procede a ejecutar el método *solicitarCrédito* del usuario, para que éste se encargue de analizar el **perfil crediticio** y la capacidad de endeudamiento. En caso de que el usuario cumpla los



requisitos, se concede el **credito** y se hacen los ajustes de saldo apropiados a la **cuenta** seleccionada al inicio del proceso. Por última se registran los cambios hechos y se le notifica al usuario el éxito en la solicitud.

*Captura de pantalla:*

```
-----
# Solicitar credito
-----
- Selección del banco
-----
1. Unalombia tasa de interés del 1.6 anual
2. PooBanco tasa de interés del 2.5 anual
3. QuitaVivienda tasa de interés del 36.0 anual
-----
Seleccione a que banco corresponde la cuenta que desea inscribir: 1
-----
R / Usted selecciono Unalombia que tiene una tasa de interés de 1.6 anual
-----
- Las cuentas que tiene asociadas son:
-----
1. Cuenta 89
2. Cuenta 23
-----
Seleccione la cuenta de la que desea guardar el crédito: 1
-----
Ingrese el monto en pesos a solicitar: 3000
-----
R / Usted ingresó 3000.0
-----
Ingrese el plazo en meses del crédito: 12
|
-----
R / Usted ingresó 12
-----
R / Saldo antes del credito 10000,0
-----
R / Tu solicitud de crédito ha sido aprobada y tu saldo actual es: 13000,0
-----
```

## Ver historial de transacciones

*Funcionamiento:*

Mediante esta funcionalidad se obtiene el reporte de todas las transacciones realizadas con una cuenta desde el momento de su creación.

Se relaciona con las clases **Usuario**, **Cuenta** y sus hijas, **Transaccion**.

*Secuencia:*

En base a los atributos del **usuario** se despliega en la interfaz su lista de **cuentas** asociadas para que se seleccione la cuenta de la que se quiere ver el historial. Posteriormente se accede a los atributos de dicha **cuenta** para extraer cada **transacción** que ha realizado y así, finalmente, desplegar un texto formateado que describa cada transacción.

*Captura de pantalla:*

```
-----
# Ver Historial
-----
- Las cuentas que teiene asociadas son:
-----
1. Cuenta 89
2. Cuenta 23
-----
Seleccione la cuenta de la que quiere ver el historial: 1
-----|
R / Usted eligio: 89
-----
Transferencia de 10 desde la cuenta 89 a Unalombia por valor de 650.00006.
Transferencia de 10 desde la cuenta 89 al usuario con CC: 98 con cuenta 26 por valor de 9250.0.
-----
```

## Transferir

*Funcionamiento:*

Con esta funcionalidad se realizan transferencias entre dos cuentas o directamente a un banco en el caso de querer pagar un crédito. Ya que el diseño es interbancario, **NO** es necesario que las transferencias sean entre el mismo banco.

Se ve relacionada con las clases **Usuario**, **Cuenta** y sus hijas, **Banco**, **Transaccion** y **Credito**, **Gestor (interface)**.

*Secuencia:*

El **usuario** debe seleccionar la **cuenta de origen** desde la que realizará la transferencia y el tipo de transferencia que hará, de donde se desprenden dos secuencias:

1. Si selecciona pagar **crédito** y cuenta con dicho crédito, se usarán los atributos para verificar la información y notificar al usuario sobre las condiciones de la operación. Si éste decide continuar se creará una **transacción** y mediante los métodos de la **interface** se harán los ajustes a la cuenta origen. Finalmente se hacen los registros pertinentes.
2. En caso de que seleccione transferencia a otra cuenta, se le preguntará si quiere transferir a cuentas inscritas o no inscritas. Si es a no inscritas se le solicitará los datos de identificación de la cuenta destino, e internamente con los datos ingresados se extraerá de **banco** el objeto **cuenta** correspondiente a la cuenta de destino. Tras

esto se realizan las verificaciones para hacer los ajustes de saldo en cada cuenta que describen la **transaccion** y finalmente se deja registro en las cuentas involucradas.

#### Captura de pantalla: Transferencia - A otra cuenta

```
-----
# Transferir
-----
- Selecciona el producto de origen que quieres usar:
-----
1. Cuenta 26
2. Cuenta 69
-----
Seleccione la cuenta de la cual quiere realizar una transferencia: 2
-----
R / Usted eligio la cuenta 69
-----
- Selección tipo de transferencia
-----
Selecciona qué tipo de transferencia quieres hacer
1. Pagar credito
2. Transferencia a otra cuenta
-----
Ingrese el tipo de transferencia: 2
-----
R / Seleccionaste: tranferir a otra cuenta.
-----
- Seleccione a qué tipo de cuenta quieres transferir:
-----
1. Cuentas inscritas
2. Cuentas no inscritas
-----
Ingrese aca la selección: 2
-----
Seleccionaste: cuentas no inscritas.
-----
- Selección del banco
1. Unalombia
2. PooBanco
3. QuitaVivienda
-----
Seleccione de que banco es la cuenta que desea inscribir:
1
-----
hasta aca voy bien
Usted selecciono Unalombia
-----
Ingrese el número de cuenta de destino: 26|
-----
El número de la cuenta ingresado es: 26
-----
Ingrese el valor a transferir - Cuentas no inscritas: 984000
-----
El valor a transferir ingresado: 984000.0
-----
Envio exitoso. El saldo de su cuenta es de 1000.0.
-----
Quedaste con un saldo de: 1000.0
```

## Transferencia- Pago crédito

```
-----
# Transferir
-----
- Selecciona el producto de origen que quieres usar:
-----
1. Cuenta 89
2. Cuenta 23
-----
Seleccione la cuenta de la cual quiere realizar una transferencia: 1
-----
R / Usted eligio la cuenta 89
-----
- Selección tipo de transferencia
-----
Selecciona qué tipo de transferencia quieres hacer
1. Pagar credito
2. Transferencia a otra cuenta
-----
Ingrese el tipo de transferencia: 1
-----
R / Seleccionaste: pagar credito
-----
R / Tu crédito es de 3000.0 y pagarás una cuota de 650.00006.
-----
- Confirmar pago
-----
1. Si
2. Volver al menú de funcionalidades.
-----
Ingrese su respuesta: 1
|
-----
R / Tu pago ha sido exitoso. Tu credito restante es de 1700.0
-----
```

# Testeo manual de las funcionalidades

---

## Creación de algunos bancos

```
Banco Unalombia = new Banco("Unalombia", (float)1.6);  
Banco PooBanco = new Banco("PooBanco", (float)2.5);  
Banco QuitaVivienda = new Banco("QuitaVivienda", (float)36.0);
```

---

## Creación de algunos usuarios

```
Usuario juanPerez = new Usuario(null, 1000000, 10, null);  
  
Usuario hernestoPerez = new Usuario(null, 1000000, 98, null);
```

---

## Creación de algunas cuentas

```
Cuenta cuenta1 = new Ahorro(89, juanPerez, (float)10000.0, QuitaVivienda, "ahorro",  
(float)36.0);  
  
Cuenta cuenta2 = new BajoMonto(69, hernestoPerez, (float)1000000.0, PooBanco,  
"bajoMonto", (float)5.0, (float)3000000.0, (float)3000000.0);  
  
Cuenta cuenta3 = new Corriente(23, juanPerez,  
(float)50000000.0, Unalombia, "corriente", (float)2000000);  
  
Cuenta cuenta4 = new BajoMonto(26, hernestoPerez, (float)6000000.0, Unalombia,  
"bajoMonto", (float)5000.0, (float)3000.0, (float)3000.0);
```

---

## Funcionalidad inscribir cuenta

---

- **¿Que usuario desea realizar?**

El primer usuario con **C: 10** quiere inscribir una cuenta

- **Primera selección: 1**
- **Selección de la funcionalidad: 1**

---

- **Caso 1: (El número de cuenta NO existe en el banco):**

- Seleccione de que banco es la cuenta que desea inscribir: **1**
    - **Usted seleccionó Unalombia**
  - Seleccione el tipo de cuenta: **1**
    - **Usted seleccionó bajomonto**
  - Seleccione el número de cuenta: **69**
    - **Numero de cuenta 1 no existe en el banco: Unalombia**
- 

- **Caso 2: (El número de cuenta existe en el banco):**

- Seleccione de que banco es la cuenta que desea inscribir: **1**
    - **Usted seleccionó Unalombia**
  - Seleccione el tipo de cuenta: **1**
    - **Usted seleccionó bajoMonto**
  - Seleccione el número de cuenta: **26**
  - Seleccione el número de cedula: **98**
    - **Las cuentas inscritas del usuario 10 son: 26**
- 

- **Caso 3: (El número de cuenta existe en el banco):**

- Seleccione de que banco es la cuenta que desea inscribir: **2**
    - **Usted seleccionó PooBanco**
  - Seleccione el tipo de cuenta: **1**
    - **Usted seleccionó bajomonto**
  - Seleccione el número de cuenta: **69**
  - Seleccione el número de cedula: **98**
    - **Las cuentas inscritas del usuario 10 son: 26 69**
- 

- **¿Que usuario desea usar?**

- hernestoPerez

**No tendrá cuentas inscritas**

---

## Funcionalidad romperTopes

---

- ***¿Que usuario desea realizar?***

El primer usuario con **C: 10** quiere romper topes a sus cuentas

- juanPerez
- 

- **Caso 1: (No tiene cuentas de bajo monto):**

- Esta funcionalidad no está habilitada para tus cuentas.
- 

- ***¿Que usuario desea realizar?***

El segundo usuario con **C: 98** quiere romper topes a sus cuentas

- hernestoPerez
  - **Selección de la funcionalidad: 2**
- 

- **Caso 1: (Si tiene cuentas de bajo monto):**

- - **¡Recuerde! El procedimiento de romper topes consiste en transformar su cuenta de tipo Bajo monto, a una cuenta de ahorros convencional, eliminando las limitaciones de este tipo de cuentas. Este proceso tiene un costo de 15.000 pesos que pagará una única vez.**
  - Para continuar seleccione la cuenta de bajo monto que desea transformar: **1**
    - **Tu solicitud ha sido aprobada, se descontará 15.000 de tu saldo para realizar el proceso. Espera un momento...**
    - **Tu cuenta ha sido actualizada y ahora no tiene topes.**
    - **Tu nueva cuenta de ahorros ahora tiene un saldo de: <SaldoCuentaAhorros>.**
- 

## Funcionalidad solicitarCredito

---

Nota: Puede que los pasos que se crean a continuación no se ejecuten en el sistema de manera similar si no se deslianza, esto se debe a que se tienen una comprobación aleatoria (Ver documentación, asignación de **comportamientoPago**)

---



- **¿Que usuario desea realizar?**

El segundo usuario con **C: 10** solicitar un crédito

- juanPerez
  - **Selección de la funcionalidad: 3**
- 

- **Caso 1: (Aprobación del crédito):**

**NOTA:** Todos los usuarios inician sin ningún crédito.

- Seleccione de que banco es la cuenta que desea inscribir: **1**
  - **Usted selecciono Unalombia que tiene una tasa de interés de 1.6 anual**
- Las cuentas que tiene asociadas son:
  - **1. Cuenta 89**
  - **2. Cuenta 23**

Entrada **2**

- Ingrese el monto en pesos a solicitar **3000**
    - **R / Usted ingresó 3000.0**
  - Ingrese el plazo en meses del crédito **12**
    - **Tu solicitud de crédito ha sido aprobada y tu saldo actual es: 50003000,0**
- 
- 

- **Caso 2: (Error porque ya se tiene un crédito):**

Si el usuario con **CC : 10**, intenta pedir otro crédito, se imprimirá.

- **-El usuario 10 ya tiene un crédito activo**
- 

- **¿Que usuario desea realizar?**

El segundo usuario con **C: 98** quiere solicitar un crédito

- hernestoPerez
  - **Selección de la funcionalidad: 3**
- 

- **Caso 3: (La cuota tentativa del crédito supera la capacidadDeEndeudamiento):**

- Seleccione de que banco es la cuenta que desea inscribir: **3**
  - **Usted selecciono QuitaVivienda que tiene una tasa de interés de 36.0 anual**



- **Las cuentas que tiene asociadas son:**

**1. Cuenta 26 2. Cuenta 69**

- Seleccione la cuenta de la que desea guardar el crédito: **1**
- Ingrese el monto en pesos a solicitar: **900.000**
- Ingrese el plazo en meses del crédito: **2**
- **Credito rechazado por falta de capacidad de endeudamiento**

---

## Funcionalidad transferir

---

- ***¿Que usuario desea realizar?***

El primer usuario con **C: 10** transferir

- juanPerez
- **Selección de la funcionalidad: 5**

---

- **Caso 1: (Pago de crédito):**

- **■ Selecciona el producto de origen que quieres usar:**
  - 1. Cuenta 89**
  - 2. Cuenta 23**
- Seleccione la cuenta de la cual quiere realizar una transferencia: **1**
  - Usted eligio la cuenta 89**
- Selecciona qué tipo de transferencia quieres hacer
  - 1. Pagar credito
  - 2. Transferencia a otra cuenta: **1**
- Seleccionaste: pagar credito Tu crédito es de 3000.0 y pagarás una cuota de 650.0.
  - 1. Sí
  - 2. Volver al menú de funcionalidades.

**Sí**

**- Tu pago ha sido exitoso. Tu credito restante es de 1700.0 R / Tu cuenta quedó con un saldo de 9350.0**

---

- ***¿Que usuario desea realizar?***

El segundo usuario con **C: 98** transferir

- hernestoPerez
  - **Selección de la funcionalidad: 5**
- 

- **Caso 2: (Transferencia):**

- - **Selecciona el producto de origen que quieres usar:**
    - 1. Cuenta 26**
    - 2. Cuenta 69**
- Selecciona la cuenta de la cual quiere realizar una transferencia: **2**
  - **Usted eligio la cuenta 69**
- Selecciona que tipo de transferencia quieres hacer
  - \1. Pagar credito
  - \2. Transferencia a otra cuenta:  
Entrada: **2**
- Seleccionaste: tranferir a otra cuenta.  
Selecciona a qué tipo de cuenta quieres transferir:
  - \1. Cuentas inscritas
  - \2. Cuentas no inscritas  
Entrada: **2**
- Seleccionaste: cuentas no inscritas.  
Selección del banco
  - \1. Unalombia
  - \2. PooBanco
  - \3. QuitaViviendaSelecciona de qué banco es la cuenta que desea inscribir:
  - **Entrada: 1**
  - **Usted selecciono Unalombia**
- Ingrese el número de cuenta de destino: **26**
  - **El número de la cuenta ingresado es: 26**

- Ingrese el valor a transferir - Cuentas no inscritas: **984.000**

- **El valor a transferir ingresado: 984000.0**

**Envio exitoso. El saldo de su cuenta es de 1000.0.**

**Quedaste con un saldo de: 1000.0**

---

- ***¿Que usuario desea realizar?***

El primer usuario con **C: 10** transferir a una cuenta inscrita

- juanPerez
- **Selección de la funcionalidad: 5**

---

- **Caso 3: (Transferencia a cuenta inscrita):**

- **Selecciona el producto de origen que quieres usar:**

**1. Cuenta 89**

**2. Cuenta 23**

- Seleccione la cuenta de la cual quiere realizar una transferencia: **1**
- Selecciona qué tipo de transferencia quieres hacer

**1. Pagar credito 2. Transferencia a otra cuenta**

Entrada: **2**

- Seleccione a qué tipo de cuenta quieres transferir:

**1. Cuentas inscritas 2. Cuentas no inscritas**

Entrada: **1**

- Seleccion cuenta destino **1. Cuenta 26 2. Cuenta 69**

Entrada: **1**

- **Digite el valor que desea transferir: 9000**
- **R / Valor ingresado: 9000.0**

---

## Funcionalidad ver historial

- ***¿Que usuario desea realizar?***

El primer usuario con **C: 10** quiere romper topes a sus cuentas

- juanPerez
  - Selección de funcionalidad: **4**
- 

s - Las cuentas que teiene asociadas son:

**\*\*1.Cuenta 89\*\***

**\*\*2.Cuenta 23\*\***

Entrada: **\*\*89\*\***