

Cajero Automático con POO

1. Introducción

1.1. Propósito

El propósito de este documento es definir los requisitos para el desarrollo de un cajero automático utilizando el enfoque de Programación Orientada a Objetos (POO).

1.2. Alcance

Este proyecto abarca la implementación de un cajero automático con funcionalidades básicas como retiros, ingresos, consulta de saldo, inicio de sesión, cierre de sesión, relación de cuentas (cuentas hijas y padres), descarga de logs y finalización del programa.

1.3. Definiciones, acrónimos y abreviaturas

- **POO:** Programación Orientada a Objetos
- **SRS:** Especificación de Requisitos de Software

2. Descripción general

2.1. Perspectiva del producto

El cajero automático será una aplicación desarrollada en Java que simulará las funciones básicas de un cajero automático real.

2.2. Funciones del producto

Las principales funciones del cajero automático son:

- Retiros de dinero (con validación)
- Ingresos (con límite de ingresos)
- Consulta de saldo
- Inicio de sesión (usuario y contraseña)
- Cierre de sesión
- Relación de cuentas (cuentas hijas y padres)
- Descarga de logs (movimientos de cada cuenta)
- Finalización del programa

2.3. Características del usuario

El cajero automático estará diseñado para ser utilizado por usuarios bancarios registrados.

2.4. Restricciones

- El programa debe estar desarrollado en Java.
- Se utilizarán los conceptos de POO: encapsulación, herencia y polimorfismo.
- La base de dinero se cargará desde un archivo de texto junto con los usuarios.
- Existirá un máximo de intentos para el ingreso de la contraseña.
- Las cuentas hijas no podrán enviar dinero a cuentas padres.

2.5. Suposiciones y dependencias

- Se asume que los usuarios tienen conocimientos básicos sobre el uso de cajeros automáticos.
- El programa depende de la disponibilidad de un entorno de ejecución de Java.

3. Requisitos específicos

3.1. Requisitos funcionales

3.1.1. Retiros de dinero (con validación)

El cajero automático permitirá a los usuarios realizar retiros de dinero de sus cuentas. Se implementarán validaciones para garantizar que el monto de retiro no exceda el saldo disponible.

3.1.2. Base de dinero (cargada desde un archivo de texto)

La base de dinero inicial del cajero automático se cargará desde un archivo de texto al inicio del programa. De la siguiente manera:

numeroCuenta,contrasena,saldoInicial

3.1.3. Ingresos (con límite de ingresos)

Los usuarios podrán realizar ingresos de dinero en sus cuentas, pero existirá un límite máximo de ingreso por transacción de \$1.000.000

3.1.4. Consultar saldo

Los usuarios podrán consultar el saldo disponible en sus cuentas.

3.1.5. Inicio de sesión (usuario y contraseña)

El cajero automático implementará un sistema de inicio de sesión basado en usuario y contraseña. Se definirán al menos 3 usuarios al inicio del programa. Si desea cargar más lo puede hacer añadiendo más líneas al txt.

3.1.5.3. Máximo de intentos de ingreso de contraseña

- Se establece un máximo de intentos permitidos (3) para el ingreso de la contraseña.
- Después de alcanzar el máximo de intentos fallidos, se finaliza el programa.

3.1.5.1. Cierre de sesión (permitir que el programa siga funcionando)

Después de iniciar sesión, los usuarios podrán cerrar su sesión sin finalizar el programa, permitiendo que otros usuarios puedan acceder.

3.1.6. Relación de cuentas (cuentas hijas y padres)

Se implementará una relación de cuentas padre-hija.

3.1.6.1. Enviar de cuenta padre a cuenta hija (con validación)

- Los usuarios podrán enviar dinero de la cuenta padre a una cuenta hija, siempre que se cumplan las validaciones correspondientes.

3.1.6.2. Cuenta hija no puede enviar a cuenta padre

- Las cuentas hijas no podrán enviar dinero a la cuenta padre.

3.1.7. Descarga de logs (movimientos de cada cuenta)

- Se implementará una funcionalidad para descargar los movimientos (logs) de cada cuenta.

3.1.8. Cerrar sesión sin finalizar el programa

- Los usuarios podrán cerrar su sesión sin finalizar el programa.

3.1.9. Botón para finalizar el programa

- Se incluirá una opción o botón separado para finalizar completamente el programa.
- Esta opción se manejará en la interfaz de usuario correspondiente.

3.2. Requisitos no funcionales

3.2.1. Rendimiento

El cajero automático debe responder de manera ágil y eficiente a las solicitudes de los usuarios.

3.2.2. Seguridad

Se implementarán medidas de seguridad, como la validación de usuarios y contraseñas.

3.2.3. Fiabilidad

El cajero automático debe ser confiable y capaz de manejar situaciones excepcionales sin fallos.

3.2.4. Usabilidad

La interfaz de usuario debe ser intuitiva y fácil de usar, siguiendo las mejores prácticas de diseño.

3.2.5. Mantenibilidad

El código fuente debe estar bien documentado y estructurado para facilitar futuras modificaciones y actualizaciones.

3.2.6. Portabilidad

El cajero automático debe ser ejecutable en diferentes plataformas y sistemas operativos que admitan Java.

3.3. Requisitos de implementación

3.3.1. Lenguaje de programación: Java

El cajero automático se desarrollará en Java.

3.3.2. Uso de encapsulación, herencia y polimorfismo

Se aplicarán los principios de POO, incluyendo encapsulación, herencia y polimorfismo, en el diseño e implementación del sistema.

4. Casos de uso

(Se adjunta diagrama de casos de usos en la carpeta Documentación)

5. Otros requisitos

5.1. Requisitos de documentación

Se proporcionará documentación completa en cada línea de código para mayor entendimiento por otros desarrolladores.

5.2. Requisitos de calidad

El código fuente deberá cumplir con estándares de calidad y buenas prácticas de programación.

5.3. Requisitos de prueba

Se realizarán pruebas unitarias, de integración y de sistema para garantizar el correcto funcionamiento del cajero automático.

6. Apéndices

6.1. Modelo de datos

El modelo de datos del sistema del cajero automático con POO está representado por el siguiente diagrama de clases:

(Se adjunta diagrama de clases en la carpeta Documentación)

La clase `AppCajero` es la clase principal que contiene el método `main` para iniciar la aplicación del cajero automático.

La clase `MenuCajero` es responsable de mostrar y manejar el menú principal del cajero automático. Tiene una asociación con la clase `Banco` y una composición con la clase `Cliente`. La clase `Banco` maneja la lista de clientes y el inicio de sesión. Tiene una composición con la clase `Cliente`.

La clase `Cliente` representa a un cliente del banco y contiene información como el número de cuenta, contraseña, saldo y una lista de cuentas asociadas. Tiene una composición con la clase `Cuenta`.

La clase `Cuenta` representa una cuenta bancaria y contiene información como el saldo, si es la cuenta principal o no, y el número de cuenta. La clase `CuentaHija` hereda de la clase `Cuenta` y representa una cuenta hija asociada a un cliente. Tiene una asociación con la clase `Cliente`.

Este modelo de clases cumple con los requisitos de encapsulación, herencia y polimorfismo de la POO.

6.2. Diccionario de datos

- **numeroCuenta:** Cadena de texto que representa el número de cuenta de un cliente.
- **contrasena:** Cadena de texto que representa la contraseña asociada a una cuenta.
- **saldo:** Valor numérico que representa el saldo disponible en una cuenta.
- **principal:** Valor booleano que indica si una cuenta es la cuenta principal (`true`) o una cuenta secundaria (`false`).
- **clientePadre:** Objeto de la clase `Cliente` que representa al cliente al que pertenece una cuenta hija.

6.3. Descripción de la interfaz de usuario

La interfaz de usuario del cajero automático con POO será una aplicación de línea de comandos. Al iniciar el programa, se mostrará un menú principal con las siguientes opciones:

1. Iniciar sesión
2. Salir

Si el usuario selecciona la opción 1, se le solicitará ingresar su número de cuenta y contraseña. En caso de que las credenciales sean válidas, se mostrará el menú de operaciones con las siguientes opciones:

1. Retirar dinero
2. Consultar saldo
3. Consignar dinero
4. Consignar dinero a otra cuenta
5. Descargar movimientos
6. Cerrar sesión

Si el usuario selecciona la opción 2 en el menú principal, el programa finalizará.

Las opciones del menú de operaciones funcionarán de la siguiente manera:

1. **Retirar dinero:** Se solicitará al usuario ingresar el monto a retirar. Se realizarán las validaciones correspondientes para verificar que el saldo sea suficiente.
2. **Consultar saldo:** Se mostrará el saldo actual del usuario.
3. **Consignar dinero:** Se solicitará al usuario ingresar el monto a depositar. Se realizarán las validaciones correspondientes para verificar que el monto no exceda el límite máximo permitido por transacción.
4. **Consignar dinero a otra cuenta:** Se mostrará una lista de cuentas hijas asociadas al usuario. El usuario seleccionará la cuenta de destino y luego ingresará el monto a transferir. Se realizarán las validaciones correspondientes para verificar que el saldo sea suficiente y que la cuenta de destino sea una cuenta hija válida.
5. **Descargar movimientos:** Se generará un archivo de texto con los movimientos (logs) de cada una de las cuentas del usuario, incluyendo el saldo actual.
6. **Cerrar sesión:** El usuario cerrará su sesión y volverá al menú principal, permitiendo que otros usuarios puedan acceder.

Adicionalmente, se incluirá un botón o una opción separada en el menú principal para finalizar completamente el programa.

Esta interfaz de línea de comandos permitirá al usuario interactuar con el cajero automático y realizar las operaciones básicas de manera intuitiva y fácil de usar.