**Documentación del Proyecto: Simulador de Banco**

**Presentación del Proyecto**

El Simulador de Banco es una aplicación diseñada para proporcionar una experiencia simulada de gestión financiera, permitiendo a los usuarios interactuar con cuentas bancarias virtuales y realizar diversas operaciones financieras, tales como depósitos, retiros, transferencias y consultas de saldo. Este proyecto se ha desarrollado con el objetivo de ofrecer una herramienta educativa y práctica para aquellos que deseen comprender y practicar conceptos bancarios básicos en un entorno simulado.

**Finalidad y Objetivos**

La finalidad principal del proyecto es proporcionar a los usuarios una plataforma de simulación realista que les permita familiarizarse con las operaciones bancarias cotidianas de una manera segura y educativa. Los objetivos específicos incluyen:

1. Educación Financiera: Facilitar la comprensión de conceptos bancarios básicos, como depósitos, retiros y transferencias.
2. Práctica Segura: Ofrecer a los usuarios un entorno virtual donde puedan practicar y perfeccionar sus habilidades financieras sin riesgos reales.
3. Desarrollo de Habilidades de Programación: Proporcionar un proyecto práctico para aquellos que deseen mejorar sus habilidades de programación, especialmente en C++.

**Características Clave del Proyecto**

* Gestión de Cuentas Bancarias: Creación y gestión de cuentas bancarias virtuales con funciones como depósito, retiro, transferencia y consulta de saldo.
* Registro de Transacciones: Cada cuenta mantiene un historial detallado de transacciones que permite un seguimiento exhaustivo de las operaciones realizadas.
* Seguridad: Incorpora características de seguridad como cambio de contraseña y bloqueo/desbloqueo de cuentas.
* Simulación en Tiempo Real: La aplicación simula transacciones y actualiza los saldos en tiempo real.
* Generación de Tickets: Capacidad para imprimir tickets detallados de transacciones, proporcionando un registro tangible de las operaciones realizadas.

**Instrucciones de Uso**

1. Compilación del Código: Utilice un compilador C++ compatible para compilar el código fuente.
2. Ejecución del Programa: Ejecute el programa compilado y siga las instrucciones del menú para interactuar con las funciones del simulador.
3. Operaciones Bancarias: Realice diversas operaciones como depósitos, retiros, transferencias y consultas de saldo según las opciones del menú.
4. Seguridad de la Cuenta: Cambie la contraseña de la cuenta y bloquee/desbloquee la cuenta según sea necesario.
5. Generación de Tickets: Utilice la opción correspondiente en el menú para imprimir un ticket detallado de las transacciones.

**Tecnologías Utilizadas**

El proyecto se ha desarrollado utilizando el lenguaje de programación C++. Este lenguaje fue elegido por su eficiencia, versatilidad y amplia aplicabilidad en el desarrollo de sistemas de software.

**Entorno de Desarrollo**

El desarrollo del proyecto se llevó a cabo en un entorno de desarrollo integrado (IDE) que admitiera la compilación y ejecución de código C++. Se utilizo un IDE Visual Studio 2022

**Funcionamiento del Proyecto**

**Estructura del Código**

El código del simulador de banco se organiza en clases que representan las entidades clave del sistema: Transaccion y CuentaBancaria. Estas clases están diseñadas para encapsular la lógica de las transacciones y la gestión de cuentas, respectivamente.

**Interfaz de Usuario**

La interfaz de usuario se presenta mediante un menú interactivo que permite al usuario seleccionar diversas opciones para gestionar su cuenta bancaria virtual. Cada opción del menú corresponde a una función específica, como realizar depósitos, retiros, transferencias, entre otras.

**Interfaz de Interacción del Banco:**

La interfaz de interacción del simulador de banco está diseñada para proporcionar una experiencia intuitiva y funcional al usuario. A través de un menú interactivo, el usuario puede acceder a diversas funcionalidades ofrecidas por el banco simulado. Aquí se describe la interfaz y las opciones disponibles:

**Menú Principal:**

Al ejecutar el programa, se presenta un menú principal que muestra las opciones disponibles. Cada opción está numerada y el usuario puede ingresar el número correspondiente para seleccionar una operación. Las opciones son las siguientes:

1. **Realizar Depósito:** Permite al usuario realizar un depósito en su cuenta bancaria. Se le pedirá ingresar el monto a depositar.
2. **Realizar Retiro:** Permite al usuario realizar un retiro de su cuenta bancaria. Se le pedirá ingresar el monto a retirar.
3. **Transferir Fondos:** Permite al usuario transferir fondos entre su cuenta y otra cuenta bancaria. Se le pedirá ingresar el monto y la cuenta de destino.
4. **Consultar Saldo:** Muestra al usuario el saldo actual de su cuenta bancaria.
5. **Ver Historial de Transacciones:** Presenta un historial detallado de las transacciones realizadas, incluyendo fechas, descripciones y montos.
6. **Cambiar Contraseña:** Permite al usuario cambiar la contraseña de su cuenta. Se le pedirá ingresar la contraseña antigua y la nueva.
7. **Bloquear/Desbloquear Cuenta:** Ofrece la posibilidad de bloquear o desbloquear la cuenta, según la situación actual de la misma.
8. **Información Detallada de la Cuenta:** Muestra información detallada sobre la cuenta, incluyendo el titular, saldo actual y estado de bloqueo.
9. **Realizar Múltiples Transacciones:** Permite al usuario realizar varias transacciones consecutivas sin volver al menú principal.
10. **Transferencia Programada:** Proporciona la opción de programar una transferencia de fondos para una fecha futura.
11. **Salir:** Finaliza la ejecución del programa.

**Flujo de Interacción:**

1. El usuario selecciona una opción ingresando el número correspondiente.
2. Dependiendo de la opción seleccionada, se le pedirá al usuario ingresar información adicional, como montos, contraseñas o cuentas destinatarias.
3. Se ejecuta la operación solicitada y se muestra el resultado o información correspondiente.
4. El usuario puede repetir el proceso seleccionando otra opción o salir del programa.

**Advertencias y Seguridad:**

* Se implementa seguridad en la entrada de contraseñas y se manejan posibles excepciones para garantizar una experiencia segura.
* La opción de bloquear/ desbloquear cuenta requiere la contraseña actual para evitar accesos no autorizados.

Esta interfaz busca proporcionar un entorno claro y accesible para que los usuarios interactúen con las funciones del banco simulado de manera efectiva y segura.

**Funcionamiento del Proyecto de Simulador de Banco:**

El simulador de banco implementa un conjunto de clases y funciones que modelan la gestión de cuentas bancarias y las transacciones asociadas. Aquí se detalla el funcionamiento clave del proyecto:

1. **Clase Transaccion:**

* Representa cada transacción realizada en una cuenta bancaria.
* Atributos:
  + **fecha**: Almacena la fecha y hora de la transacción.
  + **descripcion**: Descripción de la transacción (depósito, retiro, transferencia, etc.).
  + **monto**: Monto asociado a la transacción.

2. **Clase CuentaBancaria:**

* Representa una cuenta bancaria con sus operaciones y detalles asociados.
* Atributos:
  + **titular**: Nombre del titular de la cuenta.
  + **contrasena**: Contraseña de la cuenta (almacenada y manejada de forma segura).
  + **saldo**: Saldo actual de la cuenta.
  + **bloqueada**: Indica si la cuenta está bloqueada.
  + **historial**: Vector que almacena instancias de **Transaccion** para mantener un registro de actividades.
* Métodos Principales:
  + **realizarDeposito(monto)**: Realiza un depósito en la cuenta.
  + **realizarRetiro(monto)**: Realiza un retiro de la cuenta si hay fondos suficientes y la cuenta no está bloqueada.
  + **realizarTransferencia(destinatario, monto)**: Transfiere fondos a otra cuenta si hay fondos suficientes y la cuenta no está bloqueada.
  + **obtenerSaldo()**: Devuelve el saldo actual de la cuenta.
  + **obtenerHistorial()**: Devuelve el historial de transacciones.
  + **bloquearCuenta(contrasena)**: Bloquea la cuenta si la contraseña ingresada coincide.
  + **desbloquearCuenta()**: Desbloquea la cuenta.
  + **cambiarContrasena(nuevaPass, antiguaPass)**: Cambia la contraseña si la contraseña antigua coincide.
  + Métodos adicionales para facilitar la interacción y la gestión de la cuenta.

3. **Menú de Interacción (en la función main):**

* Presenta un menú interactivo al usuario con opciones numéricas.
* Cada opción corresponde a una operación en la cuenta bancaria.
* Se utiliza un bucle para permitir al usuario realizar múltiples operaciones antes de salir.

4. **Funciones Adicionales (mostrarHistorial, realizarMultiplesTransacciones, transferenciaProgramada, etc.):**

* Añaden funcionalidades específicas al menú principal para mejorar la experiencia del usuario.
* Ejemplos:
  + **mostrarHistorial**: Muestra el historial de transacciones de la cuenta.
  + **realizarMultiplesTransacciones**: Permite al usuario realizar varias transacciones consecutivas.
  + **transferenciaProgramada**: Representa la base para la implementación futura de transferencias programadas.

5. **Manejo de Excepciones:**

* Se implementa el manejo de excepciones para garantizar la robustez del programa.
* Se utilizan excepciones para casos como montos no válidos, contraseñas incorrectas, etc.

**Flujo de Ejecución:**

1. **Inicio del Programa:**
   * Se crean cuentas bancarias de ejemplo.
   * Se presenta el menú principal.
2. **Interacción con el Usuario:**
   * El usuario selecciona una opción del menú.
   * Se solicita y procesa la información adicional según la opción elegida.
3. **Ejecución de Operaciones:**
   * Se ejecuta la operación correspondiente en la cuenta bancaria.
   * Se registran las transacciones en el historial.
4. **Actualización de la Interfaz:**
   * Se muestra el resultado de la operación al usuario.
   * Se puede repetir el proceso o salir del programa.

Este diseño modular y orientado a objetos facilita la ampliación del proyecto y la incorporación de nuevas características en el futuro. Además, el manejo cuidadoso de contraseñas y el registro detallado de transacciones contribuyen a la seguridad y transparencia del simulador de banco.

**Registro de Transacciones**

El sistema mantiene un historial detallado de transacciones asociadas a cada cuenta bancaria. Este registro permite a los usuarios realizar un seguimiento de todas las operaciones realizadas, proporcionando transparencia y visibilidad.

**Seguridad y Gestión de Cuenta**

Se incorporan medidas de seguridad como la capacidad de cambiar la contraseña de la cuenta y bloquear/desbloquear la cuenta según sea necesario. Estas características añaden un nivel adicional de protección y control.

**Generación de Tickets**

La función de generación de tickets permite a los usuarios obtener un registro físico de las transacciones realizadas, ofreciendo una forma tangible de seguimiento y archivo.

**Este proyecto se presenta como una herramienta educativa y práctica, brindando a los usuarios la oportunidad de mejorar sus habilidades financieras y de programación en un entorno simulado y seguro. La documentación proporcionada aquí debería ser suficiente para compilar, ejecutar y comprender el funcionamiento del simulador de banco.**