**Descripción del código.**

El código proporcionado implementa un sencillo sistema de banca en línea, permitiendo a los usuarios registrarse, iniciar sesión, realizar depósitos y retiros, y a un administrador ver el historial de movimientos. El sistema utiliza archivos de texto para almacenar información de usuarios y movimientos.

**Estructura y Funcionalidades Principales**

* **Funciones principales:**
  + RegistrarMovimiento: Registra un movimiento (depósito, retiro, registro de usuario) en un archivo de texto.
  + RegistrarUsuario: Crea un nuevo usuario, guardando su nombre de usuario y contraseña en un archivo individual.
  + IniciarSesion: Verifica las credenciales de un usuario para permitir el acceso al sistema.
  + RetirarDinero: Permite a un usuario retirar dinero de su cuenta, con ciertas restricciones.
  + DepositarDinero: Permite a un usuario depositar dinero en su cuenta.
  + Administrador: Muestra el historial completo de movimientos registrados.
* **Flujo del programa:**
  + El programa presenta un menú principal al usuario.
  + Dependiendo de la opción seleccionada, se ejecuta la función correspondiente.
  + Las funciones de usuario (depósito, retiro) actualizan el saldo del usuario y registran el movimiento.
  + La función de administrador muestra el historial completo de movimientos.

**Documentación Detallada de las Funciones**

**RegistrarMovimiento(const string &tipo, const string &usuario, float monto = 0)**

* **Propósito:** Registra un movimiento en el archivo "movimientos.txt".
* **Parámetros:**
  + tipo: Tipo de movimiento (e.g., "Registro de usuario", "Retiro", "Depósito").
  + usuario: Nombre de usuario asociado al movimiento.
  + monto: Monto del movimiento (opcional, por defecto es 0).
* **Retorno:** Ninguno.

**RegistrarUsuario()**

* **Propósito:** Crea un nuevo usuario y lo registra en el sistema.
* **Parámetros:** Ninguno.
* **Retorno:** Ninguno.

**IniciarSesion(string &usuario)**

* **Propósito:** Verifica las credenciales de un usuario y permite el inicio de sesión.
* **Parámetros:**
  + usuario: Referencia a una cadena para almacenar el nombre de usuario ingresado.
* **Retorno:** true si el inicio de sesión es exitoso, false en caso contrario.

**RetirarDinero(const string &usuario, float saldo, int montoRetiro)**

* **Propósito:** Permite a un usuario retirar dinero de su cuenta.
* **Parámetros:**
  + usuario: Nombre de usuario.
  + saldo: Saldo actual del usuario.
  + montoRetiro: Monto a retirar.
* **Retorno:** Nuevo saldo después del retiro.

**DepositarDinero(const string &usuario, float saldo, float montoDeposito)**

* **Propósito:** Permite a un usuario depositar dinero en su cuenta.
* **Parámetros:**
  + usuario: Nombre de usuario.
  + saldo: Saldo actual del usuario.
  + montoDeposito: Monto a depositar.
* **Retorno:** Nuevo saldo después del depósito.

**Administrador()**

* **Propósito:** Muestra el historial completo de movimientos registrados.
* **Parámetros:** Ninguno.
* **Retorno:** Ninguno.

**Consideraciones y Mejoras**

* **Seguridad:**
  + La contraseña se almacena en texto plano, lo que es inseguro. Se recomienda utilizar técnicas de hash para almacenar contraseñas de forma más segura.
  + No hay validación de entrada para prevenir ataques de inyección.
* **Escalabilidad:**
  + Utilizar una base de datos relacional sería más adecuado para gestionar una mayor cantidad de usuarios y transacciones.
* **Funcionalidad:**
  + Se podrían agregar más funcionalidades como transferencias entre cuentas, consulta de historial personal, etc.
* **Interfaz de usuario:**
  + Se podría mejorar la interfaz de usuario para hacerla más amigable y intuitiva.