Manual de Usuario: Sistema Bancario

Página 1

1. Introducción

Bienvenido al Sistema Bancario 2.0. Este software simula la gestión de un banco del mundo real, combinando dos sistemas principales:

1. **El Lobby (Flujo Físico):** Gestiona la llegada de clientes, los ordena en una **cola de prioridad** y los asigna a **ventanillas** libres.
2. **La Bóveda (Flujo de Datos):** Procesa las **transacciones** (depósitos, retiros), mantiene un **historial** de operaciones y permite **deshacer** acciones.

El sistema utiliza **persistencia de datos**: toda la información de clientes y cuentas se carga desde archivos `.txt` al iniciar y se guarda automáticamente al salir.

2. Primeros Pasos y Persistencia de Datos

Para iniciar el programa:

1. Asegúrate de tener la carpeta `data/` con los archivos `clientes.txt` y `cuentas.txt` (pueden estar vacíos o con los datos de prueba).
2. Ejecuta el programa desde tu terminal: `./build/BankQueue.exe`

Flujo de Datos (¡Importante!):

- **Carga Automática:** Al ejecutar el programa, el sistema lee `data/clientes.txt` y `data/cuentas.txt`. Carga todos los clientes en la "Base de Datos" (`std::list`) y, simultáneamente, los añade a la "Fila de Espera" (`std::priority_queue`).
- **Guardado Automático:** El sistema **SOLAMENTE** guarda el estado actual del banco cuando seleccionas la **Opción 0 (Salir)**. Los archivos `.txt` se sobrescriben con la nueva información.

Advertencia: Si cierras la ventana de la terminal sin usar la Opción 0, **todos los cambios de la sesión se perderán**.

3. Opciones del Menú Principal (El Lobby)

Estas opciones gestionan el flujo físico de clientes en el banco.

1. Agregar nuevo cliente a la fila

- **Qué hace:** Inicia el proceso para crear un nuevo cliente y su cuenta bancaria asociada.
- **Interfaz:** Te pedirá secuencialmente:
    - Nombre (ej. "Juan")
    - Apellido (ej. "Pérez")
    - Edad (ej. 45)
    - Prioridad Especial (0=Automático, 1=VIP, 2=PREFERENCIAL).
        - Si eliges '0', el sistema asignará `ADULTO_MAYOR` (si edad >= 65) o `REGULAR`.
    - Saldo Inicial (ej. 500.00)
- **Resultado:** El cliente se guarda en la base de datos (`clientes.txt`), se crea su cuenta (`cuentas.txt`) y se añade **inmediatamente** a la `filaDeEspera`, reordenándose según su prioridad.

2. Procesar fila (llamar cliente)

- **Qué hace:** Es el corazón del lobby. Intenta atender al siguiente cliente.
- **Lógica:**
    1. Revisa si la `filaDeEspera` tiene clientes.
    2. Revisa si hay `ventanillasLibres`.
    3. Si ambos son verdaderos, saca al cliente con **mayor prioridad** de la fila, lo asigna a una ventanilla y abre el **Sub-Menú de Atención en Ventanilla**.
- **Resultado:** Si no hay clientes o no hay ventanillas libres, te mostrará un mensaje de error.

3. Ver estado de la fila de espera

- **Qué hace:** Muestra todos los clientes que están actualmente en la `filaDeEspera`.
- **Interfaz:** Verás una lista de los clientes **ordenados por prioridad**. Gracias al `ComparadorClientePtr`, los `VIP (1)` aparecerán en la cima y los `REGULAR (4)` al fondo. Si dos clientes tienen la misma prioridad, el que llegó primero (`horaLlegada` más antigua) aparecerá primero.

4. Ver estado de las ventanillas

- **Qué hace:** Muestra el estado actual de todas las ventanillas del banco.
- **Interfaz:** Verás una lista de cada ventanilla (ej. "Ventanilla Nro: 1") y su estado:
    - **Libre:** Esperando a un cliente.
    - **Atendiendo a: [Nombre del Cliente]:** Ocupada.

Página 2

4. Opciones del Menú Principal (La Bóveda)

Estas opciones gestionan los datos y transacciones del banco.

5. Ver historial de transacciones

- **Qué hace:** Muestra un registro de las operaciones realizadas en el banco.
- **Interfaz:** Imprime una lista (la `std::deque`) de las últimas 50 transacciones (depósitos, retiros, transferencias) que han ocurrido en el sistema, mostrando la más reciente primero.

6. Deshacer ultima transaccion

- **Qué hace:** Revierte la última operación *reversible* que se realizó.
- **Lógica:** Utiliza una **pila LIFO** (`std::stack`). Solo los **Retiros** y **Transferencias** se pueden deshacer.
    - Si la última acción fue un Retiro de $100, esta opción *depositará* $100 de vuelta en la cuenta.
    - Si la última acción fue una Transferencia, *devolverá* el dinero de la cuenta destino a la cuenta origen.
- **Resultado:** Los Depósitos no se pueden deshacer. Si la pila está vacía, te mostrará un mensaje.

7. Cargar datos de prueba

- **Qué hace:** Es un botón de **RESETEO**.
- **Lógica: Borra por completo** todos los clientes y cuentas de la memoria RAM, vacía los archivos `.txt`, y carga un conjunto predefinido de clientes (los 100 que te proporcioné).
- **Resultado:** El banco vuelve a un estado de prueba conocido.

0. Salir del sistema

- **Qué hace:** Es la **única** forma segura de cerrar el programa.
- **Lógica:** Rompe el bucle principal y llama a la función `guardarDatos()`, que sobrescribe `clientes.txt` y `cuentas.txt` con el estado actual del banco.
- **Resultado:** El programa se cierra y tus datos persisten para la próxima sesión.

5. Sub-Menú: Atención en Ventanilla

Este menú aparece automáticamente después de seleccionar la **Opción 2 (Procesar fila)**.

1. Realizar Deposito

- **Qué hace:** Añade fondos a la cuenta del cliente que está siendo atendido.
- **Lógica:** Pide un monto. Llama a `cuenta->depositar()`. Si tiene éxito, crea una `Transaccion` de tipo `DEPOSITO` y la añade al historial.

2. Realizar Retiro

- **Qué hace:** Retira fondos de la cuenta del cliente.
- **Lógica:** Pide un monto. Llama a `cuenta->retirar()`. Si el cliente tiene fondos suficientes, crea una `Transaccion` de tipo `RETIRO` y la añade al historial y a la **pila de deshacer**.

3. Realizar Transferencia

- **Qué hace:** Mueve fondos desde la cuenta del cliente actual a otra cuenta en el sistema.
- **Lógica:** Pide un monto y el número de la cuenta destino. Llama a `cuenta->transferirA()`. Si tiene éxito, crea una `Transaccion` de tipo `TRANSFERENCIA` y la añade al historial y a la **pila de deshacer**.

0. Terminar atencion

- **Qué hace:** Finaliza la interacción con el cliente actual.
- **Lógica:** Llama a `liberarVentanilla()`. El cliente "se va" y la ventanilla se añade de nuevo a la cola de `ventanillasLibres`, lista para atender a la siguiente persona.
- **Resultado:** Regresas al Menú Principal.