

BANQUE

PRESENTED BY
TOMÁS CAMILO GARCÍA LÓPEZ
GEOR SEBASTIÁN GOMEZ CORREA
EDUARDO CASTELLANOS MÁRQUEZ

- (1) PROPÓSITO
- (2) OBJETIVOS
- (3) CARACTERÍSTICAS
- (4) ALCANCE
- (5) CONCLUSIONES Y
APRENDIZAJES

EL PROPÓSITO: SIMULAR EL "LOBBY" Y LA "BÓVEDA" CONTENIDO:

- SISTEMA DE LOBBY (FLUJO FÍSICO):
 - GESTIONAR LLEGADAS DE CLIENTES EN TIEMPO REAL.
 - ORDENAR LA FILA POR UN SISTEMA DE PRIORIDAD (VIP, REGULAR, ETC.).
 - ASIGNAR CLIENTES A VENTANILLAS LIBRES DE FORMA EQUITATIVA (ROUND-ROBIN).
 - SISTEMA DE BÓVEDA (FLUJO DE DATOS):
 - GESTIONAR TRANSACCIONES (DEPÓSITOS, RETIROS).
 - MANTENER UN HISTORIAL DE OPERACIONES.
 - IMPLEMENTAR UNA FUNCIÓN DE "DESHACER".
 - GARANTIZAR LA PERSISTENCIA DE LOS DATOS AL CERRAR.
- (1)



EN EL PROYECTO SE UTILIZARON ESTRUCTURAS COMO LA COLA, EL ARRAY, LA COLA CIRCULAR Y LA PILA.

OBJETIVO Y REQUERIMIENTOS

OBJETIVO GENERAL:

UTILIZAR LAS LISTAS, COLAS Y PILAS PARA PROGRAMAR LA LÓGICA DEL SISTEMA DE COLA BANCARIO, TOMANDO EN CUENTA LA PRIORIDAD DE USUARIO, Y LA DISPONIBILIDAD DE LAS VENTANILLAS. ADEMÁS DE DESARROLLAR UN MENÚ BÁSICO DE USUARIO..

REQUERIMIENTO 1: FILA DE CLIENTES

- SOLUCIÓN: `STD::PRIORITY_QUEUE`
- POR QUÉ: ORDENAMIENTO AUTOMÁTICO ($O(\log N)$) POR PRIORIDAD Y HORA DE LLEGADA.

REQUERIMIENTO 2: FILA DE VENTANILLAS

- SOLUCIÓN: `STD::QUEUE`
- POR QUÉ: IMPLEMENTA UN CICLO FIFO ($O(1)$) PERFECTO PARA LA ASIGNACIÓN ROUND-ROBIN.

REQUERIMIENTOS:

(2)

REQUERIMIENTO 3: DESHACER TRANSACCIÓN

- SOLUCIÓN: `STD::STACK`
- POR QUÉ: LÓGICA LIFO ($O(1)$) IDEAL. ALMACENA COPIAS DE OBJETOS, NO PUNTEROS, PARA GUARDAR EL "ESTADO" EXACTO DE LA TRANSACCIÓN A REVERTIR.

REQUERIMIENTO 4: HISTORIAL ROTATIVO

- SOLUCIÓN: `STD::DEQUE`
- POR QUÉ: $O(1)$ PARA `PUSH_BACK` (AÑADIR NUEVO) Y $O(1)$ PARA `POP_FRONT` (BORRAR ANTIGUO SI SE SUPEREA EL LÍMITE). MÁS EFICIENTE QUE UN VECTOR.

CARACTERÍSTICAS DEL PROYECTO

LENGUAJE:

C++

PORCENTAJE DE IA:

PLANEACIÓN Y ESTRUCTURA: 90% IA.
LÍNEAS DE CÓDIGO: 40% IA

ENTORNO DE
DESARROLLO Y
REPOSITORIO

VISUAL STUDIO CODE + GIT HUB

(2)

ALCANCE Y LIMITACIONES

EL ALCANCE DE ESTE PROYECTO ES DEMOSTRAR LA APLICACIÓN PRÁCTICA Y LA EFICIENCIA DE LAS ESTRUCTURAS DE DATOS EN LA SIMULACIÓN DE UN SISTEMA DE GESTIÓN BANCARIA.

EL SISTEMA ESTÁ DISEÑADO COMO UNA PRUEBA DE CONCEPTO (POC) ACADÉMICA Y CUMPLE CON LOS SIGUIENTES OBJETIVOS FUNCIONALES:

- GESTIÓN DE CUENTAS: PERMITE LA CREACIÓN, CONSULTA, ACTUALIZACIÓN Y ELIMINACIÓN DE CUENTAS DE USUARIO.
- PROCESAMIENTO DE TRANSACCIONES: SIMULA OPERACIONES DE DEPÓSITO, RETIRO Y TRANSFERENCIA ENTRE CUENTAS.

- SEGURIDAD (CRÍTICA): EL PROYECTO CARECE DE LAS MEDIDAS DE CIBERSEGURIDAD ESENCIALES REQUERIDAS POR CUALQUIER SISTEMA FINANCIERO. NO SE IMPLEMENTARON:
 - ENcriptación de datos sensibles (contraseñas, datos personales).
- ESCALABILIDAD: EL SISTEMA FUE PROBADO CON UN VOLUMEN DE DATOS SIMULADO Y NO ESTÁ OPTIMIZADO PARA LA ALTA CONCURRENCIA O EL VOLUMEN TRANSACCIONAL MASIVO DE UN BANCO REAL.
- PERSISTENCIA DE DATOS: LA GESTIÓN DE LA BASE DE DATOS ES SIMPLIFICADA Y NO INCLUYE MECANISMOS DE RESPALDO, REPLICACIÓN O RECUPERACIÓN ANTE DESASTRES.

CONCLUSIONES Y APRENDIZAJES

DURANTE EL DESARROLLO DEL PROYECTO APRENDIMOS LECCIONES SOBRE EL MANEJO Y DISTRIBUCIÓN DEL TRABAJO EN EQUIPO, APRENDIMOS MUCHO DE LA SINTAXIS DE C++ Y SU USO.

APRENDIMOS A UTILIZAR ENTORNOS COMO VS CODE Y EMPEZAMOS A UTILIZAR GIT Y GITHUB.

(5)

NUESTROS ERRORES NOS PERMITIERON APRENDER A MANEJAR LA FRUSTRACIÓN PUES EL CÓDIGO NO SIEMPRE FUÉ PERFECTO, TAMBIÉN TUVIMOS LA OPORTUNIDAD DE ENTENDER LA CANTIDAD DE TIEMPO QUE ALGUNOS PROCESOS PUEDEN LLEVAR.

GRACIAS

GITHUB DEL PROYECTO

[HTTPS://GITHUB.COM/TONTOMAS/BANKQUEUE](https://github.com/tontomas/bankqueue)

E-MAIL:

TOGARCIAL@UNAL.EDU.CO,
GGOMEZCO@UNAL.EDU.CO,
EDCASTELLANOSM@UNAL.EDU.CO

PROFESOR

JONATAN GOMEZ PERDOMO

SOCIAL MEDIA:

@REALLYGREAISTE