

David Montoya - A00362450

Diego Garcia - A00362738

Planeación Tarea Integradora I

Requerimientos Funcionales:

R1: Registrar el nombre y la cédula de todos los usuarios que ingresan al establecimiento a la hora de obtener su turno.

R2: Permitir al encargado la búsqueda del usuario en la base de datos a partir de la cédula. Se le mostrará al encargado en pantalla: nombre, cedula, cuenta bancaria, tarjetas de débito/crédito, fecha de pago de la tarjeta de crédito y fecha en que se incorporó al banco.

R3: Modificar el monto de la cuenta de ahorros del usuario al solicitar un retiro o consignación.

R4: Borrar los datos de la base de datos del cliente y los incorpora a una exclusiva para aquellos que desertan de dicho banco. Asimismo, se guardará tanto la fecha como el motivo de cancelación.

R5: Pagar el monto utilizado con la tarjeta de crédito hasta el momento. Puede realizar el pago en efectivo o a través de su cuenta de ahorros.

R6: Desplegar la información de todos los usuarios en una tabla tipo “hoja de cálculo”, la cual podrá ser ordenada en 4 parámetros distintos los cuales son: Cedula, Nombre, Monto de cuenta de ahorros y fecha.

R7: Tener la capacidad de deshacer una acción.

Requerimientos No Funcionales:

RNF1: El programa utilizará 2 estructuras “queue”, una representando a los clientes registrados del banco y otra representando a los demás.

RNF2: Los usuarios del banco estarán almacenados en una estructura de datos ArrayList.

RNF3: Para la opción de deshacer se guardará el último objeto modificado en una pila para poder deshacer dicha acción, ya sea, modificar una cuenta, un retiro, borrar una cuenta, o registrar un usuario.

Método de la ingeniería:

FASE 1: Identificación del problema

Identificación de necesidades y síntomas:

- El banco requiere un software que registre los usuarios que acceden a sus servicios.
- No existe un programa que registre los usuarios que usan los servicios.
- El programa debe garantizar certeza en sus registros y accesos a la base de datos.
- El programa debe ser lo suficientemente eficiente para que los usuarios puedan llevar a cabo sus acciones correctamente.

Definición del problema

Un Banco requiere un programa para gestionar el funcionamiento de atención al cliente y su base de datos, en su sede con mayor flujo de clientes.

FASE 2: RECOPIACIÓN DE LA INFORMACIÓN NECESARIA

Para un mejor entendimiento del problema, pasamos a definir aquellos conceptos que hacen parte del funcionamiento del banco.

Cuenta Bancaria: Una **cuenta bancaria** es un contrato financiero con una entidad bancaria en virtud del cual, se registran el balance y los subsiguientes movimientos de dinero del cliente.

Tarjeta de Crédito: Una tarjeta de crédito es un medio de pago que te permite hacer compras y cancelar el valor posteriormente. Es “de crédito” porque la suma de dinero que usas cuando haces una compra, corresponde a un préstamo que te otorga la entidad financiera.

Monto: Suma total de varias partidas

Cuenta de ahorros: Una cuenta de ahorros es un producto financiero ofrecido por el Banco que te permite ahorrar tu dinero de forma segura. Tradicionalmente, la cuenta de ahorros es el producto más usado para que puedas ahorrar y disponer de tu dinero de forma rápida.

Consignación bancaria: La **consignación bancaria** consiste en la colocación de fondos en un banco u otras instituciones financieras para su custodia. Estas

David Montoya - A00362450

Diego Garcia - A00362738

consignaciones se realizan al depositar en cuentas tales como cuentas de ahorro, cuentas corrientes y cuentas del mercado monetario.

Fuentes:

- <https://www.grupobancolombia.com/personas/aprender-es-facil/como-manejar-dinero/endeudamiento-responsable/que-es-una-tarjeta-credito>
- <https://es.thefreedictionary.com/monto>
- <https://www.scotiabankcolpatria.com/personas/cuentas-e-inversion/mas-informacion/definicion-cuenta-ahorro>
- <https://www.lifeder.com/consignacion-bancaria/>

FASE 3: BÚSQUEDA DE SOLUCIONES CREATIVAS

Alternativa 1:

Usar la estructura “queue” para las 2 filas de clientes que se van a realizar. Guardar a los clientes en un “ArrayList”. Usar un “Stack” para la acción undo y usar JavaFX como herramienta para el diseño de la interfaz gráfica. Como métodos de ordenamiento de clientes se van a utilizar Merge Sort, Tim Sort, Heap Sort y Insertion Sort.

Alternativa 2:

Usar la estructura “queue” para las 2 filas de clientes que se van a realizar. Guardar a los clientes en un “LinkedList”. Usar un “Array” para la acción undo y usar JavaFX como herramienta para el diseño de la interfaz gráfica. Como métodos de ordenamiento de clientes se van a utilizar Bubble Sort, Selection Sort, y Shell Sort.

FASE 4: TRANSICIÓN DE LA FORMULACIÓN DE IDEAS A LOS DISEÑOS PRELIMINARES

Luego de deliberación, decidimos que la alternativa 1 es más factible y la seleccionamos para que pase al siguiente punto del desarrollo. Las razones por las cuales decidimos utilizar la alternativa 1 son que para la alternativa 2:

- En la implementación de ordenamientos resulta más complicado el uso de LinkedList.
- El uso de un array para guardar las acciones que se van a buscar para el undo es ineficiente.
- La complejidad temporal de los algoritmos supera el límite colocado por el banco.

FASE 5: EVALUACIÓN Y SELECCIÓN DE LA MEJOR SOLUCIÓN

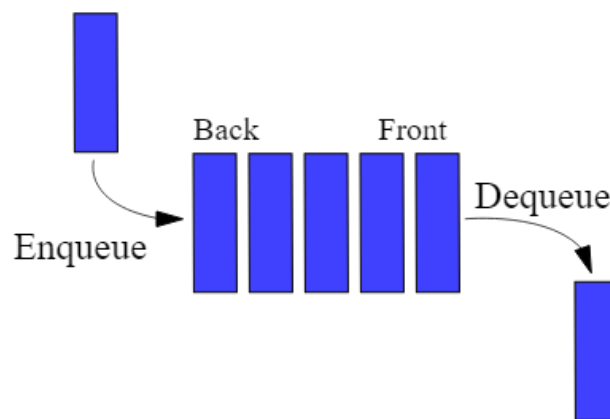
Para empezar vamos a analizar la complejidad temporal que registran los métodos de ordenamiento de la alternativa elegida.

Todos los siguientes datos están reflejados para el peor caso.

Alternativa 1:

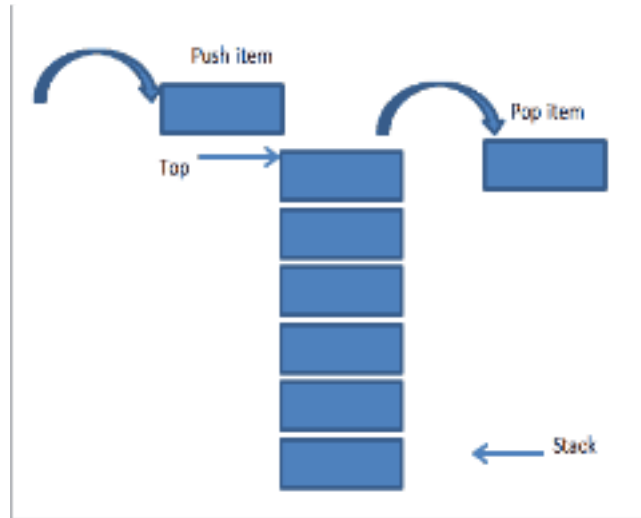
Merge Sort	$O(n * \log(n))$	Óptimo
Tim Sort	$O(n * \log(n))$	Óptimo
Heap Sort	$O(n * \log(n))$	Óptimo
Insertion Sort	$O(n^2)$	Ineficiente

A continuación, vamos a analizar porque el uso de la estructura de datos “queue” es la mejor opción para la solución del problema.



Esta estructura de datos nos permite gestionar perfectamente la atención de los clientes por su naturaleza, ya que es una estructura que nos permite tener un puntero al primer elemento en ser agregado, es decir, el primer cliente en llegar será el primero en ser atendido, y a medida que estos van llegando al establecimiento se van agregando a la cola. En conclusión, tenemos un excelente control del orden de llegada de los clientes.

El uso de una “pila” para que el usuario pueda deshacer su última acción, es una opción óptima por la siguiente razón.



La pila, al contrario de la cola, lo que nos permite es tener un apuntador al último elemento agregado, por lo cual, cada vez que se realice una modificación de los datos de un cliente, se almacena el estado anterior de este, para poder obtener de vuelta este estado en el caso de que el usuario lo requiera.