



## FASE 1: IDENTIFICACIÓN DEL PROBLEMA

### Definición del problema

Una sucursal bancaria con gran flujo de usuarios requiere el desarrollo de un módulo de software que permita gestionar las operaciones e información de los clientes y usuarios.

### Identificación de necesidades y síntomas

- La sucursal requiere poder gestionar la recepción y atención de sus usuarios utilizando dos filas: para usuarios normales y para usuarios con prioridad
- La solución al problema debe ser eficiente para que el servicio pueda mejorar sin importar la cantidad de clientes y usuarios que tenga la sucursal
- El acceso efectivo a su base de datos y la realización de optima de sus operaciones financieras.
- Almacenar clientes (personas con una cuenta en el banco) y personas ya no asociadas al banco (personas que han eliminado su cuenta)
- Permitir al cliente realizar acciones de depósito, cancelación de cuenta y pago.
- Deshacer la última acción realizada por el usuario
- Ver el estado de la fila de los usuarios con prioridad y sin prioridad (usuarios normales)
- Ver la información de lo usuario que está siendo actualmente atendido y del próximo usuario en espera de ser atendido
- Ver la lista de los clientes bancarios
- Ordenar la lista de clientes bancarios

## FASE 2: RECOPIACIÓN DE LA INFORMACIÓN NECESARIA

### Requerimientos para la solución del problema

- Buscar un cliente por el parámetro de elección
- Eliminar cuenta bancaria de un cliente por el parámetro de elección
- Deshacer la última acción financiera en caso de que exista un error al realizarla
- Crear cuenta bancaria para un cliente con los respectivos datos y pagar tarjeta de crédito del respectivo cliente.

### Restricciones para la solución del problema

- Implementar interfaz gráfica con información de los clientes del banco a través de una tabla de datos.
- Implementación de métodos de ordenamiento, tres métodos de ordenamiento eficiente y uno que sea regular.
- Implementar interfaz gráfica que represente el estado actual de las filas de usuarios.

### Definiciones

#### *Sucursal bancaria*

Una **sucursal bancaria** es un lugar donde una institución financiera ofrece asistencia personal y automatizada a los clientes. Tradicionalmente, las sucursales bancarias ofrecen depósitos, retiros, cambio de divisas, servicios de asesoría financiera, ventas de seguros y cajeros automáticos.

#### *Módulo Caja*

Este es un módulo operacional que se encarga de gestionar las funcionalidades los usuarios, tales como registrar a un nuevo cliente, realizar operaciones sobre la cuenta de un cliente dado y efectuar pagos.

#### *Módulo Fila*

Es un módulo operacional que realiza la asignación de turnos de manera prioritaria a los usuarios que llegan a la sucursal.

#### *Usuario*

Persona que requiere un servicio financiero y posee un número de cédula y un nombre.

#### *Cliente*

Tipo de usuario que tiene una cuenta registrada en el banco, es decir, que tiene una cuenta para poder efectuar operaciones financieras.

### *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.

### *Fila*

Conjunto de personas una detrás de otra

### *Software bancario*

Los productos de software bancario permiten la gestión, supervisión y control de transacciones para instituciones financieras y bancos.

### *Tarjeta de crédito*

Una tarjeta de crédito es un documento de material plástico o metal emitido por un banco o institución especial

izada a nombre de una persona, que podrá utilizarla para efectuar compras sin tener que pagar en efectivo y pudiendo, además, llevar el pago de los productos a períodos futuros.

Las tarjetas de crédito suelen tener un límite de dinero que permite que la persona compre o consuma servicios. No obstante, la entidad emisora de la tarjeta de crédito carga al comerciante un porcentaje por este servicio y en algunos casos una cuota fija anual al tenedor.

### *Pago de tarjeta de crédito*

Todo el dinero prestado se devuelve pasado un mes (a veces dos) desde el momento de la compra; en caso de que no disponga de saldo, se comenzarán a cobrar intereses como parte del préstamo.

### *Personas con discapacidades*

La discapacidad es aquella condición bajo la cual ciertas personas presentan alguna deficiencia física, intelectual o sensorial que a largo plazo afectan la forma de interactuar y participar plenamente en la sociedad

### *Cuenta de ahorros*

La cuenta de ahorro es un depósito de dinero a la vista con disponibilidad inmediata. Se caracteriza por estar instrumentalizado en una libreta en la que se recogen todas las operaciones que se realizan.

### *Tabla Hash*

Una **tabla hash**, **matriz asociativa**, **hashing**, **mapa hash**, **tabla de dispersión** o **tabla fragmentada** es una estructura de datos que asocia *llaves* o *claves* con *valores*. La operación principal que soporta de manera eficiente es la *búsqueda*: permite el acceso a los elementos (teléfono y dirección, por ejemplo) almacenados a partir de una clave generada (usando el nombre o número de cuenta, por ejemplo). Funciona transformando la clave con una **función hash** en un *hash*, un número que identifica la posición (*casilla* o *cubeta*) donde la tabla hash localiza el valor deseado.

## Stack

Una **pila** (**stack** en inglés) es una lista ordenada o estructura de datos que permite almacenar y recuperar datos, siendo el modo de acceso a sus elementos de tipo LIFO (del inglés *Last In, First Out*, «último en entrar, primero en salir»). Esta estructura se aplica en multitud de supuestos en el área de informática debido a su simplicidad y capacidad de dar respuesta a numerosos procesos.

Para el manejo de los datos cuenta con dos operaciones básicas: **apilar** (**push**), que coloca un objeto en la pila, y su operación inversa, **retirar** (o desapilar, **pop**), que retira el último elemento apilado.

## Array

Un array es un tipo de dato estructurado que permite almacenar un conjunto de datos homogéneo, es decir, todos ellos del mismo tipo y relacionados. Estos elementos pueden ser de tipo simple como caracteres, entero o real, o compuesto como String, listas, estructuras...

A los datos almacenados en un array se les denomina elementos, y al número de elementos se le denomina tamaño de un Array

## ArrayList

La clase ArrayList en Java es una clase que permite almacenar datos en memoria de forma similar a los Arrays, con la ventaja de que el número de elementos que almacena, lo hace de forma dinámica, es decir, no es necesario declarar su tamaño como pasa con los Arrays. Por lo tanto, el ArrayList es una estructura más compleja que el Array, pero a su vez mucho más potente ya que nos permite un número ilimitado de elementos sin importarnos cualquier error tipo "IndexOutOfBounds".

## Queue

Una cola (también llamada fila) es una estructura de datos, caracterizada por ser una secuencia de elementos en la que la operación de inserción *push* se realiza por un extremo y la operación de extracción *pull* por el otro. También se le llama estructura FIFO (del inglés *First In First Out*), debido a que el primer elemento en entrar será también el primero en salir.

## Árbol binario de búsqueda

**Árbol Binario de Búsqueda** (ABB). Son árboles binarios en los que se cumple que, para cada nodo, el valor de la clave de la raíz del subárbol izquierdo es menor que el valor de la clave del nodo y que el valor de la clave raíz del subárbol derecho es mayor que el valor de la clave del nodo. Las operaciones para un ABB son similares que las realizadas en un árbol binario general

## LinkedList

En ciencias de la computación, una **lista enlazada** es una de las estructuras de datos fundamentales, y puede ser usada para implementar otras estructuras de datos. Consiste en una secuencia de nodos, en los que se guardan campos de datos arbitrarios y una o dos referencias, enlaces o punteros al nodo anterior o posterior. El principal beneficio de las listas enlazadas respecto a los vectores convencionales es que el orden de los elementos enlazados puede ser diferente al orden de almacenamiento en la memoria o el disco, permitiendo que el orden de recorrido de la lista sea diferente al de almacenamiento.

### *HeapSort*

Este algoritmo consiste en almacenar todos los elementos del vector a ordenar en un montículo (*heap*), y luego extraer el nodo que queda como nodo raíz del montículo (cima) en sucesivas iteraciones obteniendo el conjunto ordenado. Basa su funcionamiento en una propiedad de los montículos, por la cual, la cima contiene siempre el menor elemento (o el mayor, según se haya definido el montículo) de todos los almacenados en él. El algoritmo, después de cada extracción, recoloca en el nodo raíz o cima, la última hoja por la derecha del último nivel. Lo cual destruye la propiedad heap del árbol. Pero, a continuación, realiza un proceso de "descenso" del número insertado de forma que se elige a cada movimiento el mayor de sus dos hijos, con el que se intercambia. Este intercambio, realizado sucesivamente "hunde" el nodo en el árbol restaurando la propiedad montículo del árbol y dejando paso a la siguiente extracción del nodo raíz.

### *Insertion Sort*

Esta es la idea detrás del ordenamiento por inserción. Itera sobre las posiciones en el arreglo, comenzando con el índice 1. Cada nueva posición es como la nueva carta que tomas del mazo, y necesitas insertarla en el sitio correcto en el subarreglo ordenado a la izquierda de esa posición.

## **FASE 3: BÚSQUEDA DE SOLUCIONES CREATIVAS**

Para este paso, pensamos soluciones propias dado que es un problema dirigido a un sector muy específico con características especiales y no se encuentran soluciones en fuentes bibliográficas. Para esto realizamos una lluvia de ideas en la que cada miembro del equipo daba una solución desde su perspectiva

### *Solución asignación de turnos*

Se propone modelar la asignación de turnos mediante características especiales de personas que requieren una atención prioritaria. Dichas filas se modelarían mediante estructuras de datos regida por los conceptos LIFO y FIFO, Stacks y Queue específicamente. Se implementará dos filas, una sin restricción alguna y otra para recibir a las personas en condición especial.

### *Solución a la gestión de la información de clientes.*

Se implementaría el modelamiento de los clientes mediante distintas estructuras de datos con características especiales que permitan una inserción, búsqueda, ordenamiento modificación y eliminación de los mismos, las cuales podrían ser Hash Table, Heaps, BST, LinkedList, ArrayList, entre otros.

### *Solución consulta de información de los clientes*

Se propone modelar una tabla tipo hoja de cálculo donde se visualice la información de los usuarios ordenando por medio de algoritmos como Insertion, QuickSort, MergeSort, BucketSort, BinaryTree Sort los parámetros de datos por nombre, cédula, tiempo de vinculación del cliente y monto de dinero, brindándonos mayor eficiencia al momento de consultar la información de los clientes por medio de la interfaz

### Opciones de estructuras de datos

- BST
- Queue
- Stack
- LinkedList
- Hash Table
- Heaps
- ArrayList
- Array

### Opciones de implementaciones de ordenamiento

- MergeSort
- Quick Sort
- Insertion Sort
- Bucket Sort
- HeapSort
- Direct Hash Sorting
- BinaryTree Sort

### **Definición de las entidades del mundo del problema:**

#### *Usuario*

Representación de una persona que requiere un servicio bancario, dicha persona posee un número único de identificación y un nombre.

#### *Cliente*

Representación de un usuario que está registrado en la base de datos del banco, el cual posee los mismos atributos del usuario y, además, una cuenta bancaria, tarjeta de crédito y débito, fecha de pago de la tarjeta crédito y fecha en la que se incorporó al banco.

#### *Modulo caja*

Representación de los comportamientos del cliente tales como agregar un cliente a la fila, realiza operaciones bancarias del cliente tales como realizar un pago, crear una cuenta bancaria, eliminar

una cuenta bancaria, además conoce la persona que está siendo atendida actualmente y la siguiente que espera por atención

#### *Modulo fila*

Representación de las filas de espera, el cual contendrá un conjunto de usuarios en espera de atención. El módulo fila debe indicar el siguiente usuario a atender por su orden de llegada o prioridad de atención, además debe generar una representación gráfica del estado de las dos filas.

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

En esta fase, hemos descartado las alternativas que no son factibles. Para las estructuras de datos, hemos descartado el uso de Array y Stack, debido a que, no nos facilitan la solución del problema y resulta más conveniente utilizar otras estructuras. Además, descartamos el uso de algoritmos de ordenamiento tipo Merge Sort, Quick Sort, Bucket Sort dado que no cumplen con los requerimientos de eficiencia o no contamos con la estructura de datos necesaria para implementarlos.

#### *Alternativa 1*

Implementar las estructuras de datos Queue, BST, Hash Table, Heap y ArrayList con los ordenamientos Insertion, HeapSort, Direct Hash, BinaryTree sort

- Las estructuras de datos Hash Table, ArrayList, Queue y BST son estructuras con características dinámicas, lo que nos permite almacenar muchos datos.
- La estructuras Queue para las filas de usuarios son las que mejor representan nuestra solución
- Los algoritmos de ordenamiento de esta alternativa cumplen con los requerimientos del problema respecto a su eficiencia

#### *Alternativa 2*

Implementar estructuras de datos Queue, BST, Stack, Array y Hash Table con los ordenamientos Insertion, Direct Hash, BinaryTree Sort

- Las estructuras de datos Hash Table, Queue y BST son estructuras con características dinámicas, lo que nos permite almacenar muchos datos. Sin embargo, el Array no es una estructura de datos y solo nos permite manejar una cantidad predefinida de datos
- La estructura Stack no representa de la mejor manera nuestra solución para las filas de los usuarios
- Los algoritmos de ordenamiento no cumplen con los requerimientos dado que faltarían algunos otros algoritmos que cumplan con la eficiencia que se necesita

### **FASE 5: EVALUACIÓN Y SELECCIÓN DE LA MEJOR SOLUCIÓN**

#### *Criterios estructuración de los datos*

- Criterio A: por efectividad en su complejidad temporal al momento de realizar operaciones sobre las mismas estructuras de datos
- Criterio B: por su eficiencia al momento guardar la cantidad de clientes que necesitáramos

- Criterio C: por la eficiencia al permitirnos simular una fila de clientes que se ordenaran según las condiciones especiales establecidas

#### **Criterios implementación de ordenamientos**

- Criterio A: por la eficiencia en su complejidad temporal al momento de realizar los ordenamientos
- Criterio B: por el tipo de estructura utilizada, si no se está implementando la estructura a la que pertenece el ordenamiento no se podrá utilizar.

#### **Selección de alternativa para estructuración de los datos**

Se optó por implementar todas las alternativas, menos la tercera de la siguiente manera: Se realizó el particionamiento de todos los clientes en cuatro subgrupos, los cuales se definieron con base a cuatro rangos de sus respectivos números de identificación. No se realiza la evaluación de los criterios anteriores dado que el problema nos permite elegir todas las opciones anteriores

#### **Selección de alternativas para implementación de ordenamientos**

De acuerdo a los criterios anteriores, se concluyó que se implementarían las alternativas de ordenamiento Insertion Sort, Heap Sort, Direct Hash Sorting y BinaryTree Sort.

	Criterio A	Criterio B	Total
Merge Sort	4	3	7
Quick Sort	2	3	5
Insertion Sort	3	5	8
Bucket Sort	2	1	3
Heap Sort	5	5	10
Direct Hash Sorting	5	5	10
BinaryTree Sort	5	5	10



## Webgrafía

- Definición de fila tomada de: <https://es.thefreedictionary.com/fila>
- Definición de sucursal bancaria tomada de: [https://es.wikipedia.org/wiki/Sucursal\\_bancaria](https://es.wikipedia.org/wiki/Sucursal_bancaria)
- Definición de cuenta bancaria tomada de: [https://es.wikipedia.org/wiki/Cuenta\\_bancaria](https://es.wikipedia.org/wiki/Cuenta_bancaria)
- Definición de persona con discapacidad tomada de: <https://es.wikipedia.org/wiki/Discapacidad>
- Definición de tarjeta de crédito tomada de: <https://economipedia.com/definiciones/tarjeta-de-credito.html>
- Definición de pago de tarjeta de crédito tomada de: <https://economipedia.com/definiciones/tarjeta-de-credito.html>
- Definición Tabla Hash tomada de: [https://es.wikipedia.org/wiki/Tabla\\_hash](https://es.wikipedia.org/wiki/Tabla_hash)
- Definición Stack tomada de: [https://es.wikipedia.org/wiki/Pila\\_%28inform%C3%A1tica%29](https://es.wikipedia.org/wiki/Pila_%28inform%C3%A1tica%29)
- Definición Queue tomada de: [https://es.wikipedia.org/wiki/Cola\\_%28inform%C3%A1tica%29](https://es.wikipedia.org/wiki/Cola_%28inform%C3%A1tica%29)
- Definición de Array tomada de: <https://luismoralescotoli.wordpress.com/2016/10/01/definicion-y-utilizacion-de-arraylist-clase-string-tokenizer/>
- Definición de ArrayList tomada de: <https://luismoralescotoli.wordpress.com/2016/10/01/definicion-y-utilizacion-de-arraylist-clase-string-tokenizer/>
- Definición de Árbol binario de búsqueda tomada de: [https://www.ecured.cu/%C3%81rbol\\_binario\\_de\\_b%C3%BAqueda](https://www.ecured.cu/%C3%81rbol_binario_de_b%C3%BAqueda)
- Definición de Lista Enlazada tomada de: [https://es.wikipedia.org/wiki/Lista\\_enlazada](https://es.wikipedia.org/wiki/Lista_enlazada)
- Definición de HeapSort tomada de: <https://es.wikipedia.org/wiki/Heapsort>
- Definición de Inserción tomada de: <https://es.khanacademy.org/computing/computer-science/algorithms/insertion-sort/a/insertion-sort>