

## Introducción

La contingencia nacional y mundial han hecho que los bancos jueguen un papel preponderante en la vida del ciudadano común. Es por esto que han decidido mejorar sus sistemas informáticos, y así tener una administración más ordenada y eficaz.

Después de los resultados obtenidos por sus algoritmos de ordenamiento en el contexto de redes ópticas, usted ha sido contratado como experto en estructura de datos y algoritmos para diseñar e implementar un nuevo sistema para el manejo de la información de sus clientes, que además cumple con las 3 B.

El banco recibe la información de sus clientes en un archivo de texto separado por **tabuladores** (archivo data.txt), por lo que el programa debe ser capaz de leer este archivo, y cargar su información **inicial** en la estructura de datos que se muestra en la Figura 1.

La Figura 1 muestra una tabla de hash. Para evitar colisiones, cada *bucket* corresponde a un puntero a una lista que contiene a los clientes cuyo apellido comienza con la letra respectiva al bucket de la tabla de Hash. Es así como los clientes cuyo apellido comienza con A se encuentran en la lista del primer *bucket*, los clientes con apellidos que comienzan en B se encuentran en la lista del segundo *bucket* y así sucesivamente.

La Figura 2 muestra la estructura de un cliente junto a su lista de transacciones. Este cliente estaría en la posición “b”, dado que su apellido comienza con esa letra. Además, cada cliente tiene una lista de transacciones que siguen una estructura de *stack*: La transacción más antigua se encuentra en la posición de más abajo, mientras que la más nueva se encuentra en la posición de más arriba (justo donde apunta el puntero que viene desde la estructura del cliente). La estructura de las transacciones contiene dos números: El primero es el código de la transacción (101 para un depósito, y 201 si es un retiro).

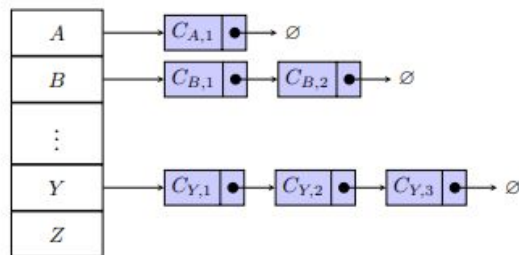


Figura 1: Estructura de datos de los clientes

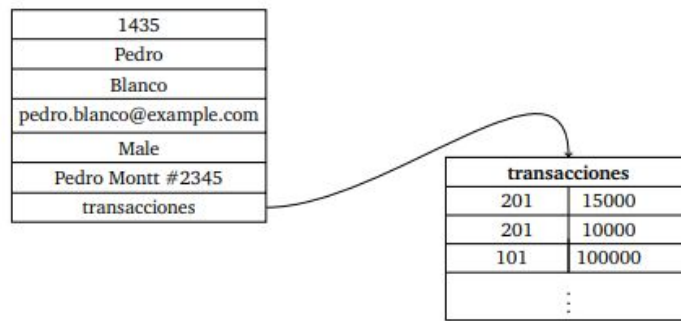


Figura 2: Estructura de datos de cada cliente, y de las transacciones

## 1. ¿Qué se espera de esta tarea?

Se espera que usted sea capaz de trabajar con una entrada de datos previamente formateada, estructurando la información dentro de un programa en C en tablas de hash, listas y pilas. Además se pretende que pueda comprender de mejor manera el funcionamiento de las estructuras de datos antes mencionadas.

## 2. Comportamiento

El programa a entregar debe ser capaz de administrar las cuentas de un banco. Por lo mismo debe contener un menú donde el usuario (trabajador del banco) seleccione entre diferentes posibilidades:

1. Cargar archivo de clientes
2. Agregar cliente
3. Eliminar cliente del sistema. Para poder eliminar un cliente del sistema, este debe tener su cuenta en cero (suma de depósitos y retiros igual a 0).
4. Buscar cliente por rut, nombre o apellido y mostrar su información.
5. Agregar transacción
6. Deshacer última transacción
7. Generar una lista de clientes que hayan depositado más de  $X$  dinero en su cuenta en **la última transacción**.
8. Generar una lista con los clientes que tienen más dinero en el banco (diferencia entre depósitos y retiros).

## 3. Archivos y contenido esperado

En esta tarea se trabajarán tres estructuras diferentes, por lo que se esperan los seis archivos correspondientes, además del main y README:

- *pila.h* y *pila.c*
  - Estructura con la definición de un elemento de la pila
  - Como mínimo las siguientes funciones:
    - *crear\_pila*
    - *agregar\_elemento\_pila*

```

Administrador de personas. Banco EDA:
=====
1.- Cargar archivo de clientes
2.- Agregar cliente
3.- Eliminar cliente del sistema
4.- Buscar cliente por rut, nombre o apellido
5.- Agregar transacción
6.- Deshacer última transacción
7.- Generar una lista de clientes según última transacción
8.- Generar una lista con los clientes que tienen más dinero en el banco

```

Figura 3: Ejemplo de menú

- `remover_elemento_pila`
- `lista.h` y `lista.c`
  - Estructura con la definición de un elemento de la lista
  - Como mínimo las siguientes funciones:
    - `crear_lista`
    - `agregar_elemento_lista`
    - `remover_elemento_lista`
- `hash.h` y `hash.c`
  - Estructura con la definición de un elemento de la tabla de hash
  - Como mínimo las siguientes funciones:
    - `crear_hash`
    - `agregar_elemento_hash`
    - `remover_elemento_hash`
- `main.c` Programa principal, con el menú que permita: Agregar un cliente, quitar un cliente, editar un cliente, hacer una transacción, editar una transacción
- `README.md` : Un archivo que contenga la información del programa, así como los datos del alumno que lo hizo.

#### 4. Plazos y últimas consideraciones

- La tarea deberá ser subida a webcursos antes del día **1 de Junio del 2020 a las 23:59 hrs.**
- Se debe subir **un archivo zip con los 8 archivos** arriba mencionados.
- Es libre de agregar las funciones que desee dentro de los 7 archivos mencionados.
- Su tarea debe compilar. En caso contrario la nota máxima a la que aspira es un 4.0.
- Queda prohibido cualquier paso de información referente a la tarea entre los estudiantes. Si se da el caso, los estudiantes que participen en tal hecho recibirán la nota mínima