

“Algoritmos y Estructuras de Datos”



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

TRABAJO PRÁCTICO

❖ **TEMA: Archivos**

❖ **PROFESOR: Ing. Pablo Mendez**

❖ **CURSO: K1004**

❖ **ENTREGA: 16/09**

Nombre	Apellido	Legajo	Correo
Lucas	Cáceres	1728830	Lucasdcaceres00@gmail.com
Verónica	Arias	1386128	Arias_veronica@live.com.ar
Ariel	Surco	1754865	Arielsurcoa@gmail.com

2020

**ALGORITMOS Y ESTRUCTURAS DE DATOS****División de Tareas**

Al inicio del proyecto nuestro equipo decidió definir los parámetros y la finalidad de cada función. Luego nos dividimos las funciones más simples para realizarlas asincrónicamente, y acordamos un horario para juntarnos en un meet y resolver las funciones más complejas.

En las juntadas sincrónicas por meet se utilizó la página de collabedit.com.

Para compartir el .cpp utilizamos onedrive.

Solución del problema

En base al problema planteado, decidimos dividir la solución en diferentes subprogramas, logrando así, una mejor división de tareas para trabajar en el código.

La solución empleada consta de la utilización de un array al que se cargarán todas las cuentas que se encuentren en el archivo cuentas.bic, por medio de la función "Levantar", con la intención de que esto facilite la implementación de los demás subprogramas, a causa de la utilización de array el proyecto queda limitado a la capacidad de memoria RAM y no al espacio de almacenamiento.

Utilizamos un total de 10 subprogramas, sin contar el menú, estos se presentan y explican a continuación:

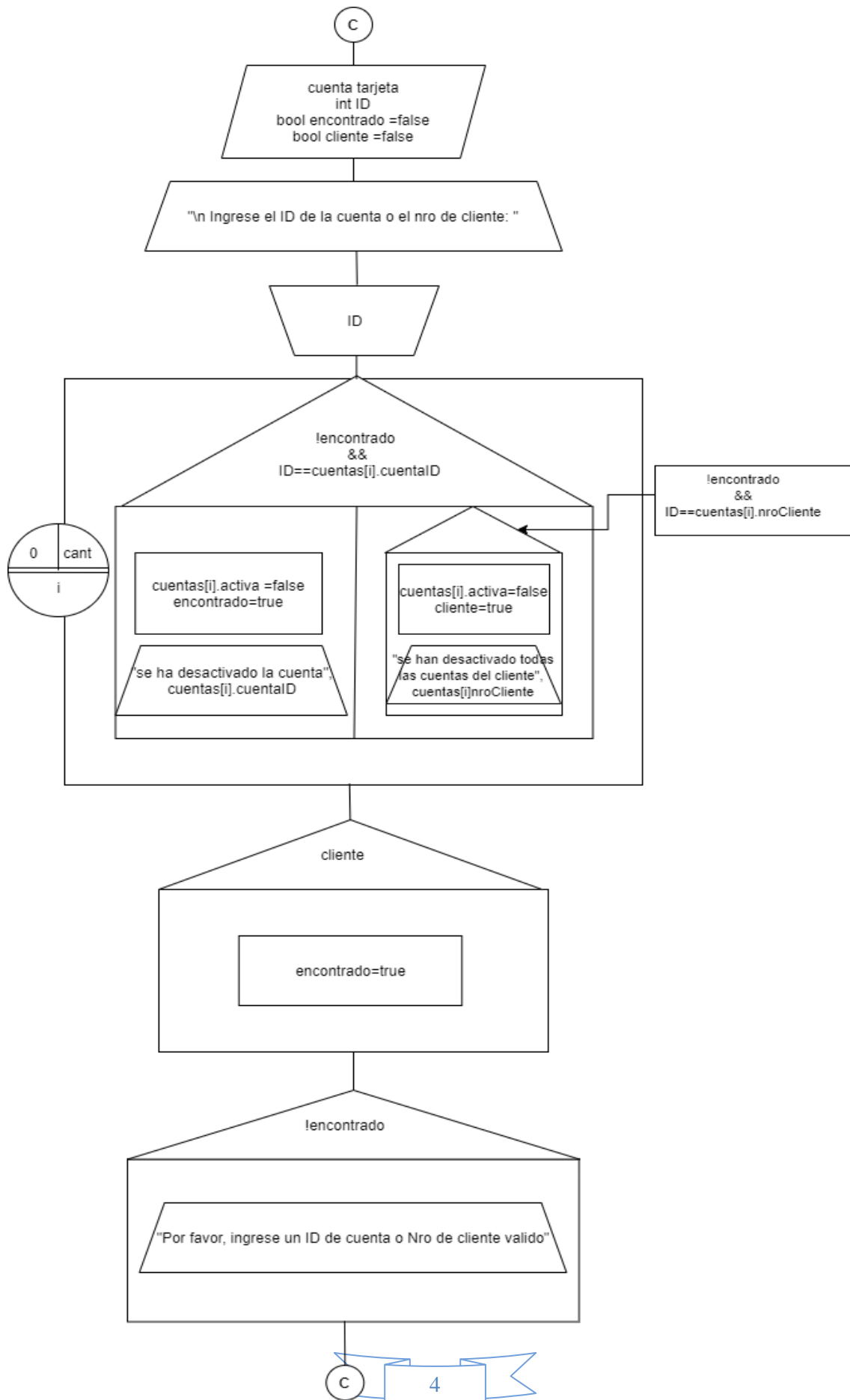
- **Levantar:** Se encarga de escribir en el array, todas las cuentas que se encuentran en el archivo "cuentas.bic".
- **Cant_Cuentas:** Recorre el archivo para devolver la cantidad de cuentas existentes.
- **Cant_Movimientos:** Recorre el archivo de movimientos y devuelve la cantidad de elementos que contiene este mismo.
- **Cargar_Cuenta:** Permite agregar una cuenta al array en el que se cargaron las cuentas con la función "Levantar".
- **Estado_Cuenta:** Permite activar o desactivar una cuenta cambiando la variable "activa" de "True" a "False" (Llámesse borrado lógico).

**ALGORITMOS Y ESTRUCTURAS DE DATOS**

- **Id_Search:** Busca una cuenta por cuentaID o por el número de cliente y muestra el contenido de la cuenta buscada.
- **Active_List:** Se encarga de filtrar las cuentas que están activas en el array de cuentas, y las pasa a un array auxiliar que se ordena en base al saldo de cada cuenta, de forma descendente, e imprime los datos de todas las cuentas activas ordenadas de la forma mencionada.
- **Lote_Proceso:** Esta función procesa el lote de movimiento, modificando los montos según el id de cuenta que éste tiene. En el caso que se modifique exitosamente se imprime la leyenda “***** **Procesamiento del lote exitoso** *****”, en el caso que el id de la cuenta en el movimiento no coincida con ninguna cuenta, se imprime la leyenda “**Se encontraron movimientos de cuentas inexistentes**”.
- **Fin_Jornada:** Sobrescribe todo el archivo “cuentas.bic” con el array actualizado, es decir, con las cuentas agregadas que estén activas y, según corresponda, con el saldo actualizado por la función “Lote_Proceso”.

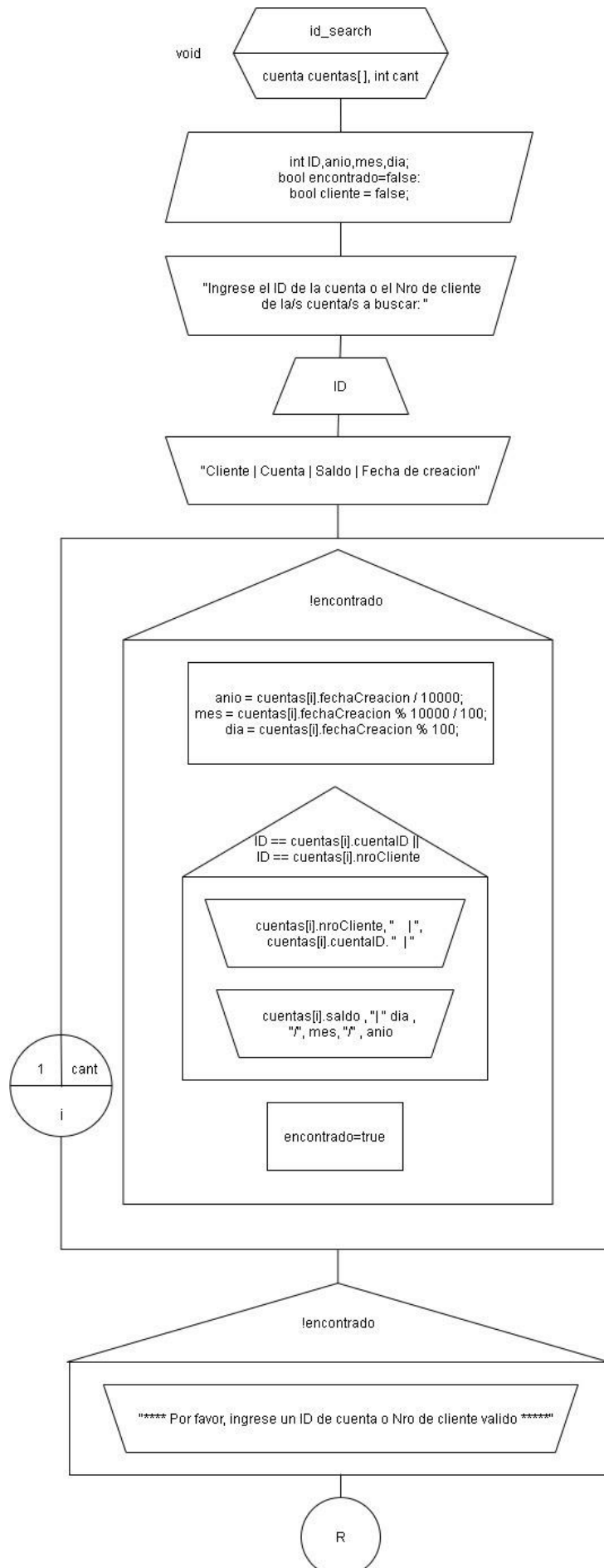


ALGORITMOS Y ESTRUCTURAS DE DATOS





ALGORITMOS Y ESTRUCTURAS DE DATOS





ALGORITMOS Y ESTRUCTURAS DE DATOS

