



Estructura de Datos y Algoritmos

Primer Semestre 2020

Tarea 2: “Manejo de TDA”

Objetivos

Desarrollar un programa en C/C++ que permita resolver el problema utilizando adecuadamente las operaciones del TDA de Lista.

Grupos de Trabajo

El trabajo es individual.

Actividades

Considere una empresa dedicada a la venta de productos puerta a puerta, donde el gerente utiliza un TDA lista para mantener la información relevante sobre sus *vendedores* y los *clientes* asociados a cada vendedor. Por cada *vendedor* se almacenan los siguientes datos: Número de cuenta bancaria, Rut, nombre, apellido, teléfono, dirección, edad, profesión, deuda y los datos de los *clientes* asociados (Vea la sección de ejemplo para clarificar las estructuras que debe utilizar). Por cada *cliente* es necesario guardar: nombre, apellido, rut, teléfono, dirección, edad, profesión, deuda y fecha de cobro.

Además considere que inicialmente se tienen guardados ciertos datos en un archivo binario llamado *ventas.txt* donde primero se almacena la cabecera de la lista de *vendedores* y luego los nodos de la lista, tenga en cuenta que luego de cada nodo de tipo *vendedor* se almacena una cabecera de la lista *clientes* con sus respectivos nodos a continuación (Vea la sección de ejemplo).

Según lo anterior, el programa debe cumplir con los siguientes requerimientos:

- Leer adecuadamente la información desde el archivo binario *ventas.txt* e ingresarlos a la lista.
- Eliminar vendedor de la lista por rut.
- Buscar vendedor por rut.
- Buscar cliente por rut.
- Mostrar la información de todos los clientes de un determinado vendedor.
- Calcular la el monto total que debe cobrar el vendedor.

Nota: Para mayor simplicidad, considere el rut sin dígito verificador.

AYUDA: Téngase en cuenta que al modificar el contenido que almacenan los *nodos*, será necesario modificar algunas operaciones del TDA LISTA para cumplir con lo solicitado (por ejemplo: es necesario modificar la operación *localiza*).

Entrega

Envíe un e-mail a matthias.clein@ufrontera.cl con Asunto: “Tarea 2 – ApellidoEstudiante” y adjunte todos los archivos necesarios para su correcto funcionamiento.

Fechas

Publicación: 04 de agosto de 2020

Entrega: 28 de agosto de 2020

Observación

El envío de la tarea puede realizarse hasta las 23:59 hrs. del día establecido para la entrega. Se descontará 1 punto por día a las tareas enviadas fuera de plazo. **La copia o plagio será sancionada con un 1.0.**

Consultas

Cualquier consulta la puede realizar a través de Slack en el canal de “consultas” o al mail matthias.clein@ufrontera.cl.

Ejemplo:

```
struct Cliente{
    char nombre[15];
    char apellido[15];
    char rut[10];
    long telefono;
    char direccion[50];
    int edad;
    char profesion[30];
    long deuda;
    char fechaCobro[11];
};
```

```
struct cabeceraCliente{
    int largo;
    nodoCliente* primero;
};
```

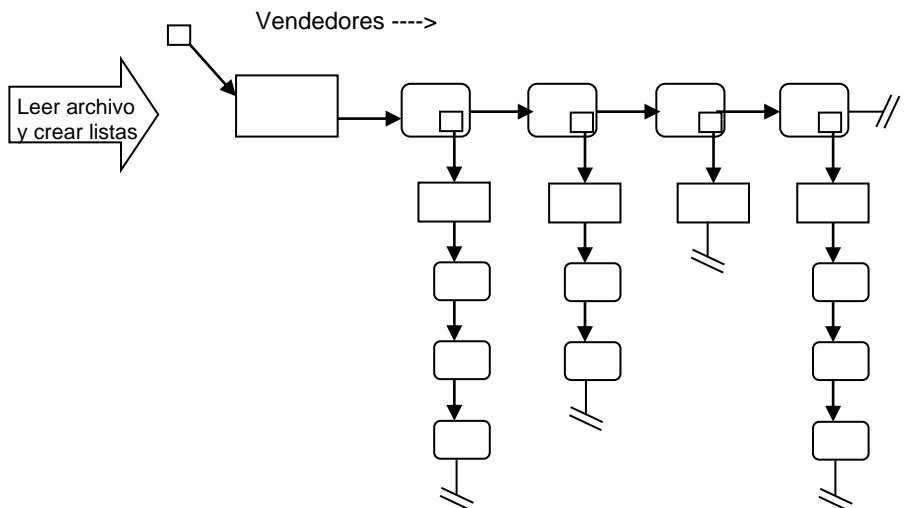
```
struct nodoCliente{
    Cliente dato;
    nodoCliente* sgte;
};
```

```
struct Vendedor{
    long cuenta;
    char nombre[15];
    char apellido[15];
    char rut[10];
    long telefono;
    char direccion[50];
    int edad;
    char profesion[30];
    long deuda;
    cabeceraCliente* clientes;
};
```

```
struct cabeceraVendedor{
    int largo;
    nodoVendedor* primero;
};
```

```
struct nodoVendedor{
    Vendedor dato;
    nodoVendedor* sgte;
};
```

ventas.txt





Estructura de Datos y Algoritmos

Primer Semestre 2020

Encabezado archivo main

```
/* Universidad de La Frontera
 * Ingeniería Civil Telemática
 *
 * IIE-344 Estructura de Datos y Algoritmos
 *
 * Tarea 2
 *
 * Autor: Nombre (mail)
 *
 * Fecha:
 *
 * Descripción general del programa:
 */
```

Encabezado funciones

```
/*
 * Nombre de la función: suma.
 * Tipo de función: int.
 * Parámetros: "a" de tipo int.
 *             "b" de tipo int.
 * Dato de retorno: "resultado" de tipo int;
 * Descripción de la función: La función suma() pide dos parámetros de tipo
 *                             entero para sumarlos y devolver el resultado
 *                             al programa principal a través de la variable
 *                             resultado.
 */
```