

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Trabajo Práctico N° 7

TEMA: Tipos de datos derivados estructuras. Estructuras anidadas. Arreglo de estructuras

CONSIGNA DE TRABAJO



Trabaje de forma ordenada en cada ejercicio. Identifique en cada caso entre la información que se brinda los datos que conforman la estructura principal, el tipo de cada uno y la presencia de estructuras que se puedan anidar. Al trabajar con módulos (funciones/procedimientos), use pasaje de parámetros por valor del tipo struct.

1. Calentando Motores

Diseñe un programa en C para gestionar el registro de proyectos y la asignación mensual de horas trabajadas por los empleados en cada uno durante un año completo. Cada proyecto debe tener la siguiente información:

- Nombre del proyecto
- Duración estimada del proyecto en meses.
- Horas trabajadas por el empleado cada mes (array de 12 elementos).
- Datos del empleado asignado:
 - Nombre
 - Puesto (Desarrollador Frontend o Desarrollador Backend)
- Fecha de inicio del proyecto: Día, Mes y Año.



Recuerde que al crear una estructura en C se está definiendo un **nuevo tipo de dato compuesto** por lo tanto puede declarar variables de este tipo.

Declare como estructura principal a **tProyecto** y como estructuras anidadas **tEmpleado** y **tFecha**. Además, implemente los siguientes módulos:

- Cargar datos:** módulo que permite ingresar la información completa **de un proyecto**.
 - Controlar que la duración estimada en meses sea un dato válido sino pedir nuevamente. Mínimo 1 mes, máximo 12 meses.
 - Para la carga de horas trabajadas por el empleado cada mes, tener en cuenta la duración del proyecto.
 - Controlar que la fecha de inicio para cualquier proyecto sea en el mes de enero del 2024. De no ser así, informar "Fecha inválida" y solicitar nuevamente.
- Mostrar datos:** módulo que muestre la información completa **de un proyecto**.
- Use los siguientes prototipos para los módulos:

```
tProyecto cargarDatos();  
void mostrarDatos(tProyecto dato);
```

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

2. Agencia Volá Paloma

Escriba un programa en lenguaje C para gestionar la información de una agencia de viajes. Realice la estructuración correcta de la siguiente información:

- Destino del Viaje
- Datos del Coordinador (Apellido, Nombre y DNI).
- Fecha de salida
- Fecha de regreso
- Costo del Viaje
- Cantidad de habitaciones reservadas (campo numérico)
- Lista de números de habitaciones reservadas (puntero para crear un arreglo dinámico)

Su tarea consiste en:

- Definir estructuras para almacenar los datos del coordinador y fecha: día, mes y año.
- Al cargar las fechas controlar que sean válidas: máximo de días de acuerdo al mes, el año puede ser el actual hasta el 2026.
- Para almacenar la lista de números de habitaciones trabajar con un arreglo dinámico. Es decir, debe declarar este miembro dentro de la estructura principal como un puntero.

Implemente los siguientes módulos:

- Cargar en el sistema los datos **de un viaje**.
- Mostrar la información completa **de un viaje**, incluyendo los números de habitaciones.
- Módulo para actualizar el costo **de un viaje**. Incrementar su valor en un porcentaje ingresado por pantalla.
- Módulo para controlar que una fecha es válida. Investigue de ser necesario.



Para agilizar la carga de datos puede hacer uso de la función rand para la carga de datos numéricos aleatorios, por ejemplo:

```
10000000 + rand() % 90000000; // entre 10.000.000 y 99.999.999  
(25000000 + rand() % 55000001) / 100.0; // entre 250.000,00 y 800.000.00
```



Volá más alto: Modifique el módulo de carga para simular la gestión de habitaciones, el programa deberá asignar números aleatorios en el rango de 100 a 400, controle que el número no exista previamente, de existir genere nuevamente.

El primer dígito del número indica el piso al que pertenece, el resto corresponde a la habitación. Por ejemplo, el número 291 corresponde al segundo piso, habitación 91. **Implemente un módulo** que cuente y muestre cuántas habitaciones del hotel están reservadas por piso.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

3. BiyuYa

Se busca desarrollar un programa en C para gestionar la información de los usuarios de una billetera virtual emergente. Cada usuario tiene los siguientes datos:

- Apellido
- Nombre
- DNI (Documento Nacional de Identidad)
- Fecha de nacimiento (día, mes, año)
- Contraseña de acceso
- Datos de la cuenta:
 - CVU (Clave Virtual Uniforme)
 - Saldo en la billetera
 - Historial de transacciones (reserva dinámica con tamaño por defecto).



Pensar cuál es la forma más conveniente de trabajar con el tamaño del historial de transacciones que puede variar según el cliente.

Defina una estructura principal para almacenar estos datos, y al menos dos estructuras anidadas. Inicialice la información de al menos dos usuarios con datos de prueba.

El programa debe ofrecer un pre-menú interactivo para seleccionar un perfil con billetera virtual:

Usuario 1: [Apellido1, Nombre1, DNI1]

Usuario 2: [Apellido2, Nombre2, DNI2]

...

Ingrese 0: Para finalizar.

El consultante selecciona el número correspondiente al perfil con el que desea interactuar, y luego se presenta un menú con las siguientes opciones:

- **Cambiar contraseña:** Permite ingresar una nueva contraseña.
- **Agregar fondos:** Permite agregar dinero en la billetera del perfil seleccionado.
- **Retirar fondos:** Permite retirar dinero de su billetera, verificando que haya saldo disponible.
- **Consultar saldo:** Muestra el saldo actual.
- **Información de la cuenta:** Muestra el nombre del perfil y el CVU.
- **Historial de transacciones:** Muestra el historial de transacciones en la billetera.
- **Volver al menú principal:** Permite regresar al menú principal para seleccionar otro perfil.



A fin de aportar modularidad, legibilidad y claridad al código trabajar cada opción del menú como un módulo de tipo void. Siempre tener en cuenta estos aspectos.

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Ejemplo de cómo inicializar datos de prueba

```
struct tCliente clientes[4] = {  
    {"Gonzalez", "Juan", 12345678, {15, 5, 1990}, "pass123", {"123456789",  
1000.0, NULL, 0}},  
    {"Lopez", "Maria", 87654321, {10, 8, 1985}, "pass456", {"987654321",  
1500.0, NULL, 0}},  
    {"Navarro", "Marcelo", 85651321, {19, 2, 1979}, "pass450", {"987654321",  
41500.0, NULL, 0}},  
    {"Medina", "Nazareno", 87654321, {21, 10, 1980}, "pass999", {"987654321",  
25500.0, NULL, 0}}  
};
```

Esta declaración e inicialización de datos crea un arreglo de 4 elementos de tipo `struct tCliente` llamado `clientes`. Cada elemento representa un cliente con toda su información. La disposición puede variar según cómo se defina la estructura principal y las anidadas.