



Universidad Autónoma de Baja California

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA



# Reporte de Laboratorio

**Lenguaje C**

**Investigación**

Arreglos de estructuras en C

**Alumno**

Joshua Osorio Osorio - 1293271

**Docente**

Fernando Elihonai Saucedo Lares

**20 de marzo de 2025**

# ÍNDICE

<b>Objetivo.....</b>	<b>2</b>
<b>Introducción.....</b>	<b>2</b>
Lista de materiales.....	2
<b>Desarrollo.....</b>	<b>3</b>
Calcular el perímetro de cualquier tipo de figura o polígono regular.....	3
Identificar si un número es primo.....	4
Identificar si el número capturado es múltiplo de algún número base, definir Cuál es y cuantas veces cabe en dicho número.....	4
Identificar si un número tiene raíz cuadrada exacta.....	4
<b>Conclusiones.....</b>	<b>4</b>
<b>Bibliografía.....</b>	<b>5</b>

## Objetivo

Aprenderás a comprender qué es una estructura y qué es un arreglo en el lenguaje de programación C. Además, seras capaz de declarar estructuras, arreglos y combinar ambos conceptos al crear arreglos de estructuras. Tambien sabrás cómo acceder a los datos almacenados dentro de ellos y utilizarlos de manera eficiente en programas reales.

## Introducción

### Qué es una estructura?

Una estructura en C es una estructura de agrupar diferentes tipos de datos bajoo un mismo nombre. Permite almacenar varias variables de distintos tipos dentro de un solo objeto organizado. Cada variable dentro de la estructura se llama miembro.

### Qué es un arreglo?

Un arreglo en C es una colección de elementos del mismo tipo de datos, organizados en posiciones consecutivas de memoria. Cada elemento del arreglo se accede mediante un índice numérico.

Tanto las estructuras como los arreglos son fundamentales para organizar y manejar conjuntos de datos complejos.

## Lista de materiales

- Computadora
- software: mingw
- Editor de texto.

# Desarrollo

## ¿Cómo declarar una estructura? [1][3]

Para declarar una estructura de datos, se utiliza la palabra clave struct más el nombre de la estructura, luego llaves y entre las llaves el tipo y nombre de la variable y cerramos con punto y coma, repetimos los pasos anteriores desde que declara el tipo y nombre para declarar un nuevo miembro.

```
struct + nombre {  
    // Declaración de miembros.  
}
```

Ejemplo:

```
C/C++  
struct Triangulo {  
    char nombre[100];  
    double l1, l2, l3;  
    int tipoTriangulo;  
};
```

Esto crea un tipo llamado struct Triangulo, que agrupa los datos nombre, l1,l2,l3 y tipoTriangulo.

## ¿Cómo declarar un arreglo? [2][3]

Un arreglo se declara especificando el tipo de dato, el nombre del arreglo y su tamaño entre corchetes [ ].

tipo de dato + nombre [cantidad de elementos]

Ejemplo:

```
C/C++  
int numeros[5]; // Arreglo de 5 enteros
```

Este arreglo almacena 5 números enteros.

¿Cómo declarar un arreglo de estructuras?[3]

Un arreglo de estructuras combina ambos conceptos: se crean una agrupación de estructuras.

Ejemplo:

Estructura + nombre de la estructura +nombre de la variable [Cantidad de elementos ]

```
C/C++  
struct Triangulo triangulos[3];
```

Aquí, triangulos es un arreglo que contiene 3 estructuras del tipo struct Triangulo. Cada elemento, como triangulos[0], triangulos[1] o triangulos[2], es una estructura completa con sus propios datos.

Para acceder a un miembro de una estructura dentro del arreglo:

```
C/C++  
triangulos[0].nombre = "escaleno";  
triangulos[0].l1 = 3.5;  
triangulos[0].l2 = 4.5;  
triangulos[0].l3 = 5.5;  
triangulos[0].tipoTriangulo = 3;
```

# Conclusiones

El uso de estructuras y arreglos en C es esencial para construir programas que necesiten manejar grandes cantidades de datos de forma organizada.

Una estructura permite almacenar múltiples tipos de datos relacionados, mientras que un arreglo facilita trabajar con múltiples elementos del mismo tipo.

Al combinar ambos conceptos en un arreglo de estructuras, se pueden modelar sistemas más complejos como listas de estudiantes, registros de productos, figuras geométricas y mucho más. Dominar estos temas amplía tu capacidad de crear programas ordenados, eficientes y fáciles de mantener.

# Bibliografía

- [1] J. A. Ruiz, “Arreglos unidimensionales y bidimensionales en C#,” 30-Oct-2024. [Online]. Available: <https://repository.unad.edu.co/handle/10596/64393>.
  
- [2] “The C programming language,” *Google Books*. [Online]. Available: [https://books.google.com.mx/books?hl=es&lr=&id=OpJ\\_0zpF7jIC&oi=fnd&pg=PP12&dq=estructuras+en++lenguaje+c&ots=2eK1giweLZ&sig=kpdFjHLmeEtTExNj-aUdt3FfTyk#v=onepage&q=estructuras&f=false](https://books.google.com.mx/books?hl=es&lr=&id=OpJ_0zpF7jIC&oi=fnd&pg=PP12&dq=estructuras+en++lenguaje+c&ots=2eK1giweLZ&sig=kpdFjHLmeEtTExNj-aUdt3FfTyk#v=onepage&q=estructuras&f=false).
  
- [4] “Estructuras,” *Google Docs*. [Online]. Available: <https://docs.google.com/presentation/d/12AIz4EhCCQHIXhGj3yWJs6H7ejjyOpsNWuB8et7ooXM/edit?hl=es#slide=id.p>.