

PROPUESTA 008  
Para Sistema bibliotecario

**PROFESOR:** Rodolfo Pacco Palomino

**SECCIÓN:** IZA

**GRUPO:** 2

**EMPRESA:** BIBLIOTECA

**HTO:** 3





# INTRODUCCIÓN

## NUESTRO EQUIPO:

O'Besso Enzo

Ingeniería de Ciberseguridad  
San Miguel  
U20241A775



Fredy Gamarra

Ingeniería de ciberseguridad  
San Miguel  
U20241A775

Ralph Ricapa

Ingeniería de ciberseguridad  
San Miguel  
U20241A775





# INTERPRETACION PROBLEMÁTICA

En el problema del formato **Biblioteca** nos piden lo siguiente:

El desarrollo de un correcto y optimizado sistema manejado por consola la cual se potencie según las funcionalidades básicas y necesarias para las bibliotecas municipales. Al igual que beneficiar a los lectores regulares y abrazar a aquellos que quieren prestarse un libro de dichos establecimientos fomentando así el hábito de la lectura y facilitando esta misma muchos más peruanos.

Para resolver este problema en c++, tendremos que considerar algunas variables a tomar en cuenta antes de proceder a crear el código, tales como:



- Un Sistema de prestamos
- Un Sistema de devoluciones
- Mayor efectividad del sistema
- No sobrecargar la memoria
- Modo de usuario
- Modo de administrador
- Datos de las bibliotecas
- Tipo de material de lectura
- Libros restantes

# Contenido

## EXPLICACION

Análisis de negocio  
Y requerimientos

1

2

## Diagramas de flujo

Diagrama de flujo del  
proceso principal

## CUADRO DE FUNCIONALIDADES

Funcionalidades completadas y  
las que faltaron

3

## ESTRUCTURAS USADAS EN EL PROYECTO

Estructuras y arreglos

4

# Resumen del proyecto

## MISIÓN

Crear un software de biblioteca para el uso de clientes y empleados, que conste con sistemas de préstamo, devoluciones y funciones extra que ayuden tanto a cliente como empleado

## VISIÓN

Queremos que otros desarrolladores colaboren y ayuden al crecimiento del proyecto, y que a su vez pueda ser adaptado para más bibliotecas.

## OBJETIVOS

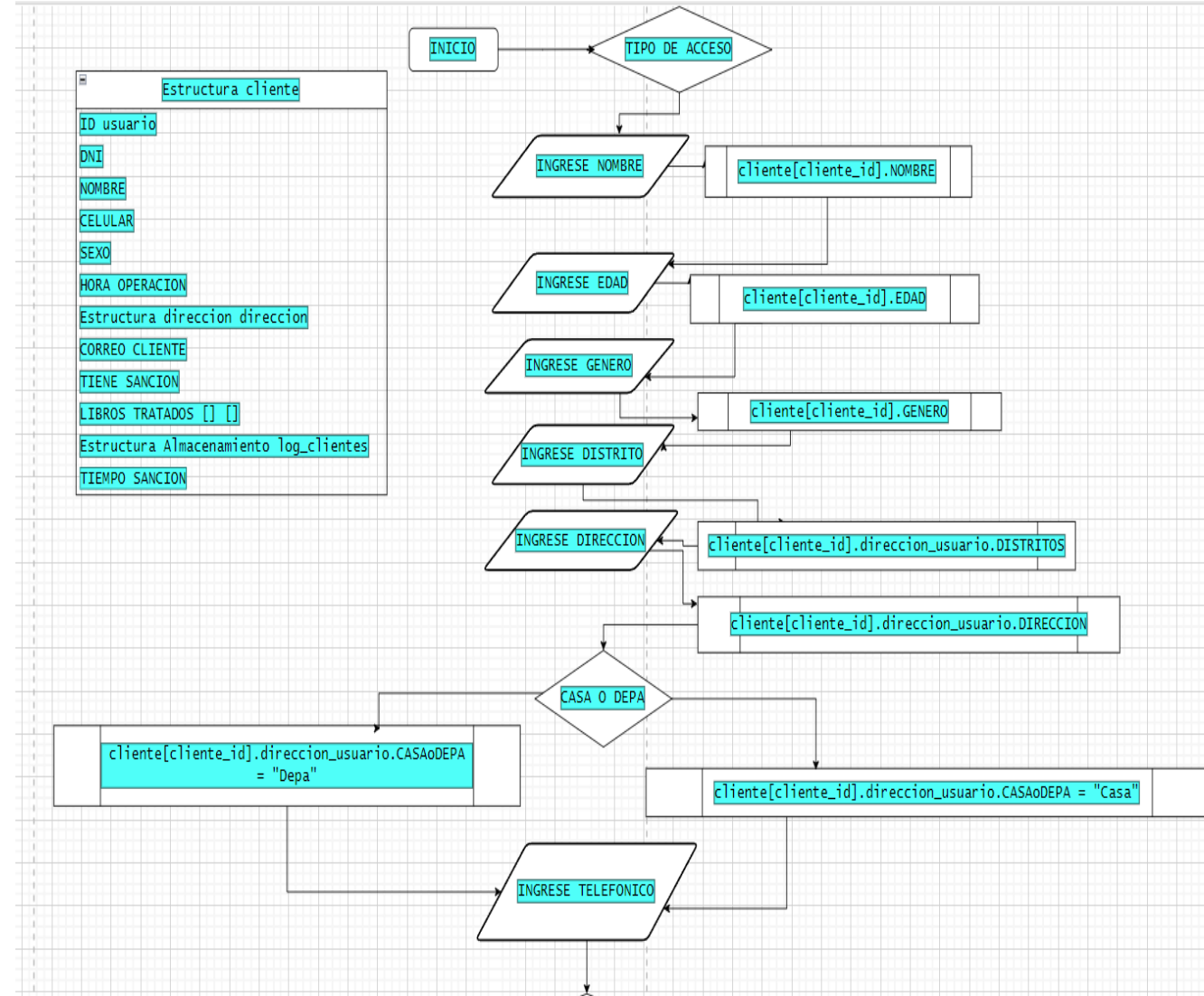
Desarrollar un sistema con funcionalidades de uso practico, fácil de adaptar y con interfaz amigable, reducir tiempos de atención y costos de personal para las bibliotecas municipales y mejorar su lado administrativo.



## DIAGRAMA DE FLUJO

### 1.- Función de registro

Aquí se registraran los usuarios para prestar o devolver un libro, hace uso de la información que se le proporcione para almacenarla y usarla luego en otros procesos.

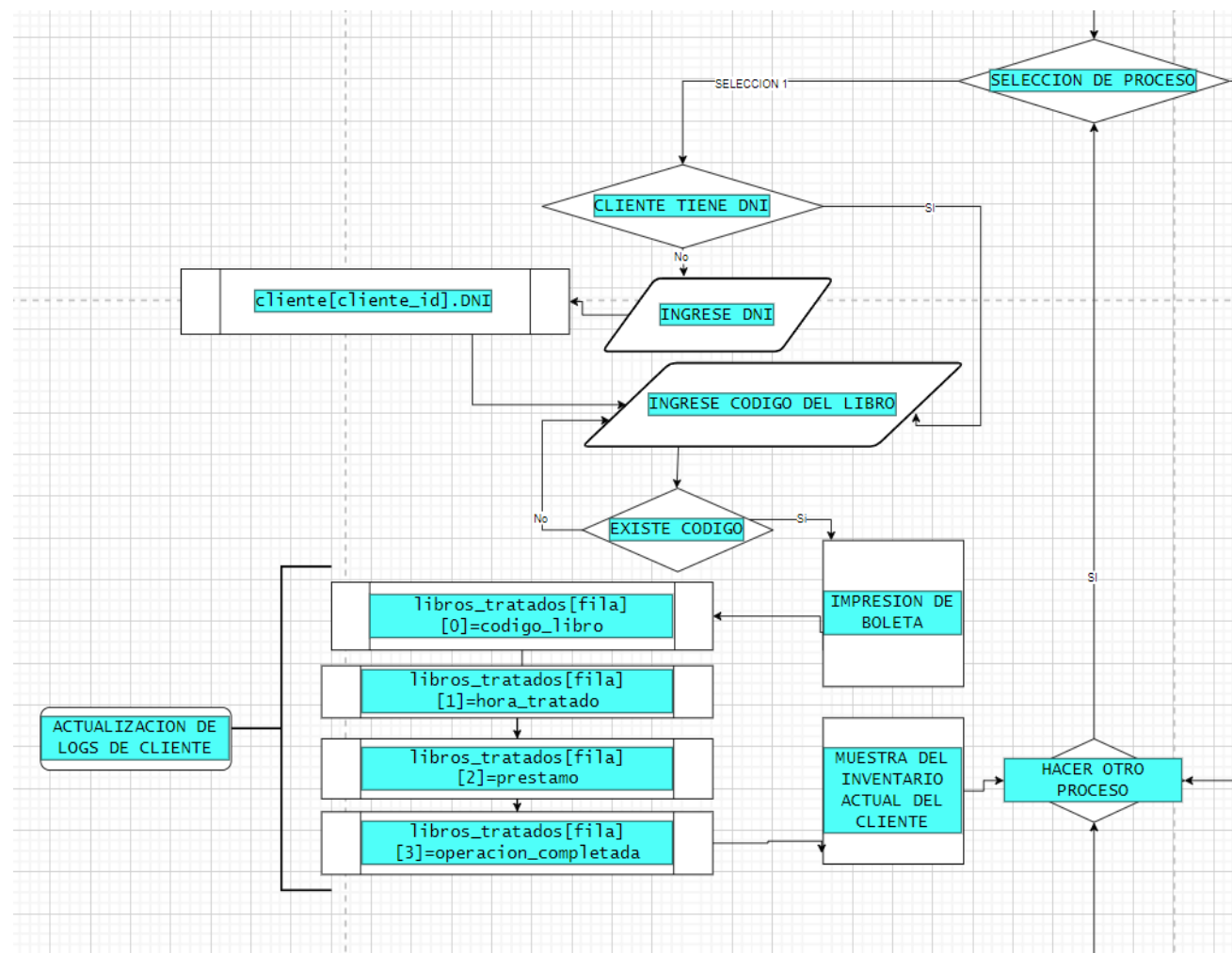




## DIAGRAMA DE FLUJO

### 2- Función de préstamo

Se solicita el **DNI** al cliente, si aún no lo ha ingresado previamente; luego se pide **el código** y tipo de la publicación prestada, seguido del día real del préstamo. Se muestra una boleta que indica si el usuario tiene alguna sanción.

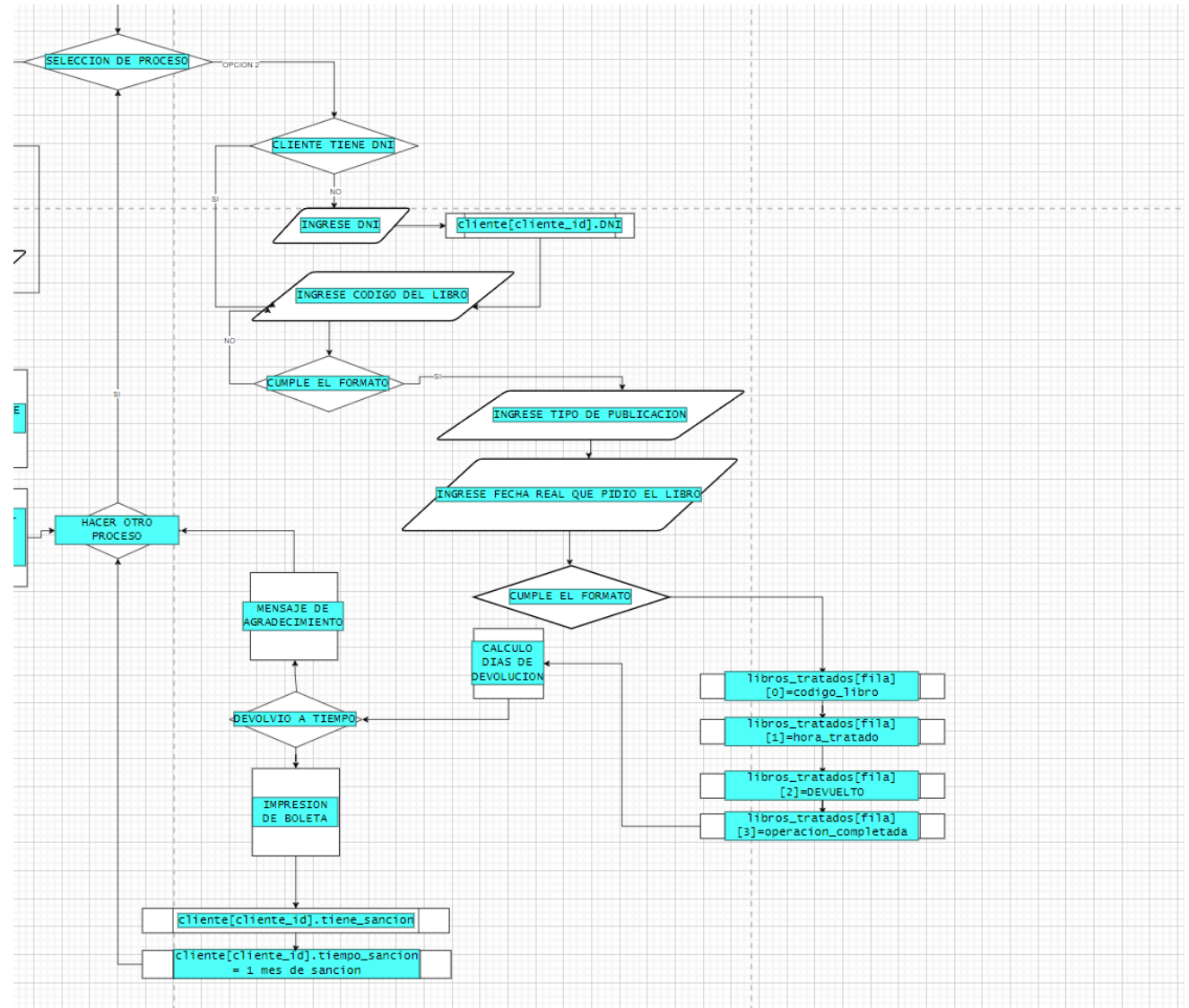


## 3.- Función devolución

En el recibo de devolución aparecen los datos del cliente y la fecha de plazo que tiene para devolver el material, además de la hora a la que se realizó la devolución del material prestado.

## ¿CUANTO TIEMPO DE SANCIONES?

La sanción será de un mes desde el día actual que devolvió la publicación. La sanción puede ser modificada por un usuario del sistema.





# Funcionalidades

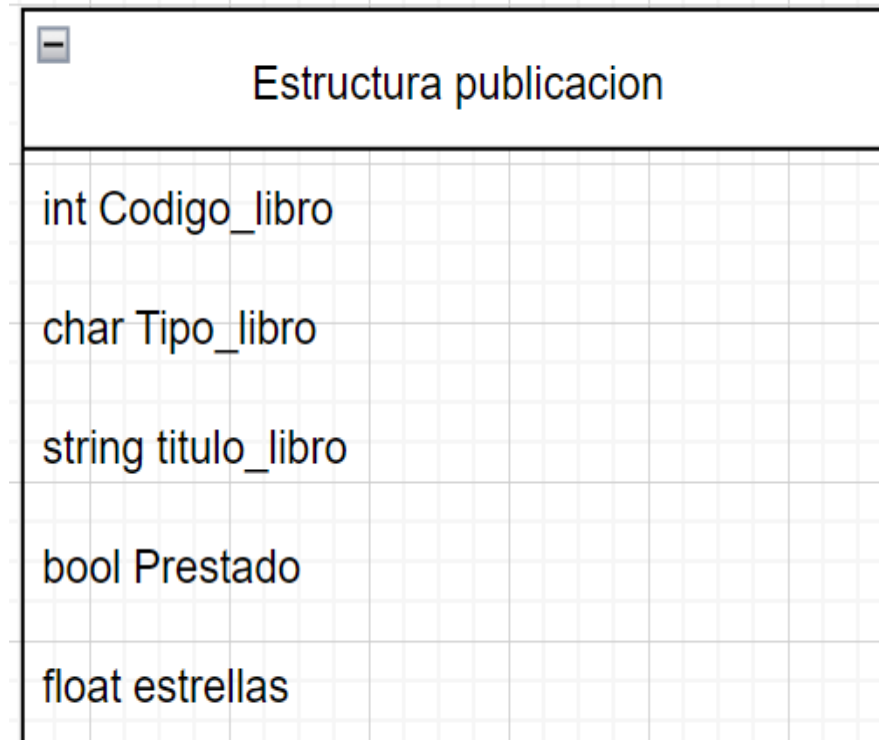
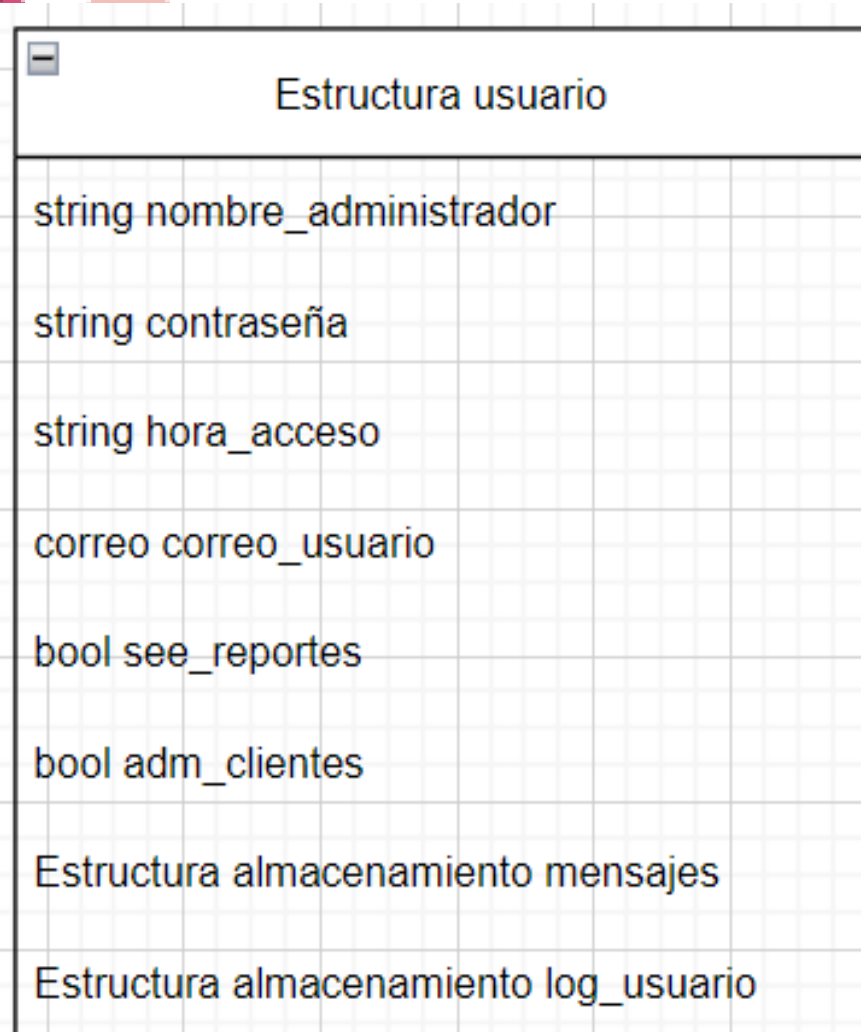
|             |  |  |  |                               |   |                 |                          |
|-------------|--|--|--|-------------------------------|---|-----------------|--------------------------|
| Completas   | <p>REPORTE DE LECTORES POR RANGO DE EDAD</p> | <p>REPORTE PUBLICACION PENDIENTE EN DEVOLVER</p> | <p>REPORTE PORCENTAJE DE TIPO DE PUBLICACIÓN</p> | <p>REPORTE GRAFICO BARRAS</p> | <p>DEVOLUCIÓN</p>                               | <p>PRESTAMO</p> | <p>REGISTRAR CLIENTE</p> |
|             | <p>ADMINISTRAR CLIENTES</p>                  |  | <p>MODIFICAR ATRIBUTOS DEL CLIENTE</p>           |                               | <p>REPORTE DE CLIENTES ACTUALES REGISTRADOS</p> |                 |                          |
| INCOMPLETAS | <p>CREAR LOGS DE CLIENTE</p>                 |  | <p>REPORTE DE LOGS DE CLIENTES</p>               |                               |   |                 |                          |

# 04

ESTRUCTURAS

## ESTRUCTURAS

| Estructura cliente                     |
|--|
| int ID usuario                         |
| string DNI                             |
| string NOMBRE                          |
| string CELULAR                         |
| char SEXO                              |
| string HORA OPERACION                  |
| Estructura direccion direccion         |
| string CORREO CLIENTE                  |
| bool TIENE SANCION                     |
| LIBROS TRATADOS [] []                  |
| Estructura Almacenamiento log_clientes |
| string TIEMPO SANCION                  |





## Estructura almacenamiento

string \*\* almacenamiento



## Estructura correo

string correo



## Estructura comentario

int comentario ID

int cliente ID

string comentario

string tiempo

## Planificar el código y hacer pruebas mientras lo escribimos

- Esto ayuda mucho a la hora del debuggin y saber que rumbo debe tomar el código.

## Crear funciones que se usen múltiples veces

- Mejora la eficacia del código, hace que sea más fácil escribir o editarlo; también hace que sea más legible.

## Preguntar a otros acerca del tema

- Preguntar a los expertos puede ayudarnos mucho a obtener información y desarrollar el código, por eso no debemos temer a preguntar.

## Usar una herramienta de confianza

- A veces los compiladores tienen diferentes formas de interpretar algunos procesos y pueden dar error.

Repositorio Público del Proyecto:

[https://github.com/Code0008/Trabajo\\_final\\_PROGRAMACION\\_BASICA-/](https://github.com/Code0008/Trabajo_final_PROGRAMACION_BASICA-/)

## Bibliografía:

- String Handling in C++. (n.d.). Obtenido de <https://repositorio.konradlorenz.edu.co/micrositios/001-1618/string.html>
- Manipulators in C++. (n.d.). Obtenido de <https://cplusplus.com/reference/iomanip/>
- Time Library in C. (n.d.). Obtenido de <https://conclase.net/c/librerias/time/ctime>
- Librerías y Cabeceras en C++. (n.d.). Obtenido de <https://es.stackoverflow.com/questions/72944/librerias-y-cabeceras-en-c#:~>
- Structures in C++. (n.d.). Recuperado de <https://learn.microsoft.com/es-es/cpp/cpp/cpp/struct-cpp?view=msvc-170>
- tm Structure in C++. (n.d.). Recuperado de <https://cplusplus.com/reference/ctime/tm/>
- C Library Documentation. (n.d.). Recuperado de <https://www.alegsa.com.ar/Diccionario/C/4149.php#gsc.tab=0>
- Generando numeros aleatorio en C++ . (n.d.). Recuperado de <https://www.azulschool.net/todos-los-grupos/grupo-de-c/forum/topic/generar-numeros-aleatorios-en-c/>
- Insertion Sort in C++. (n.d.). Recuperado de <https://www.geeksforgeeks.org/insertion-sort/>
- System Calls in C. (n.d.). Recuperado de <https://www.geeksforgeeks.org/system-call-in-c/>
- Timestamp en desarrollo web (n.d.). Recuperado de <https://desarrolloweb.com/faq/timestamp>
- Unidimensional y Multidimensionales (n.d.). Recuperado de [Arreglos Unidimensionales y Multidimensionales \(unam.mx\)](https://www.unam.mx)

# GRACIAS POR VER