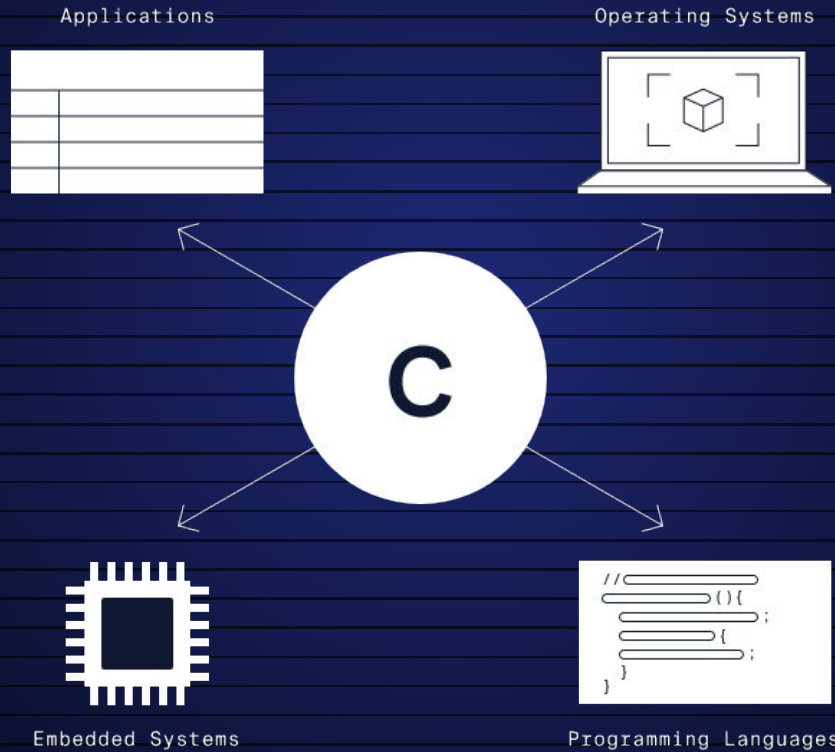


# Laboratorio de Programación

# Intro al lenguaje C

Nando Ardiles

# The C Programming Language



# El lenguaje de programación C



¡Felicitaciones! vas a aprender el lenguaje de programación C! C existe desde hace mucho tiempo y es uno de los lenguajes fundamentales de la informática. C es uno de los lenguajes más antiguos en comparación con otros lenguajes en uso y, sin embargo, sigue siendo muy popular.

La mayoría de los sistemas operativos actuales, incluido el kernel de Linux, se implementan con código C. La versión principal del lenguaje de programación Python se llama CPython porque se implementa con C. C también se ha ampliado con los lenguajes C++ y C#.

El lenguaje de programación C está en todas partes. Aprenderlo lo ayudará a convertirse en un mejor programador, listo para el próximo desafío en cualquier campo de la informática.

## Comencemos!

# Nuestro primer Programa

```
1  #include <stdio.h>
2
3  int main() {
4      // output a line
5      printf("Hello World!\n");
6      return 0;
7  }
8
```

Cuando se ejecuta este código, se muestra el siguiente texto en la terminal.

```
Hello World!
```

# Entendiendo el Código

Repasemos el código línea por línea para ver qué está pasando. No necesita entender todo de inmediato, esto es solo un primer vistazo.

- `#include <stdio.h>`: esta línea es necesaria para ejecutar la línea de código que comienza con `printf`.
- `int main(){ }`: Este es el punto de partida del código. Todo el código dentro de las llaves `{ }` se ejecuta primero.
- `// salida de una línea`: Esto es un comentario. No es una línea de código, sino un mensaje que podemos agregar al código para decirnos a nosotros mismos o a otros lo que hace el código. Cuando se ejecuta el código, esta línea se ignorará.
- `printf("¡Hello World!");`: esta línea de código imprime, o genera, el texto "¡Hola mundo!" a la consola. Imprimir texto en la consola es una forma de que un programa se comuniqué con el usuario.

## Error!: Podrías encontrar los errores en este código

```
1  #include <stdio.h>
2
3  int main() {
4      // output a line
5      printf("Hello World!\n")
6  }
```

Para Verificar el código se debe compilar "debugear" debug en inglés

# Errores de Sintaxis

Al escribir en C, debemos seguir un conjunto de reglas para que el código se ejecute correctamente. Estas reglas se conocen como sintaxis.

Veamos el código de Hello World para examinar la sintaxis común que existirá en la mayoría (si no en todos) de sus programas.

- **Case Sensitivity:** La mayoría de las palabras en el código usan todas las letras minúsculas. Esto se conoce como distinción entre mayúsculas y minúsculas. Ya sea en minúsculas o mayúsculas, ciertas palabras en el código deben seguir el caso correcto para que se ejecute el código. Las únicas líneas de texto que pueden cambiar de mayúsculas y minúsculas son el comentario y el texto entre comillas.
- **El punto y la coma:** Todas las declaraciones, como la declaración `printf()`, deben terminar con un punto y coma. Esto identifica el final de la instrucción y es necesario para que el código se ejecute correctamente.
- **Doble comillas:** El texto entre comillas dobles " se conoce como una cadena (piensa en una cadena de caracteres). Todas las cadenas deben estar entre comillas dobles.

Entonces, ¿qué sucede cuando rompemos las reglas?

La respuesta es:

## errores

El siguiente texto es un error que se genera cuando omitimos el punto y coma de la instrucción `printf()` en nuestro código Hello World.

```
script.c: In function 'main':  
script.c:6:1: error: expected ';' before '}'  
token  
}  
^
```



# Outputs

## printf()

**printf()** se conoce como una función y realiza la acción de imprimir texto en la consola. "¡Hola mundo!\n" es una cadena. Una cadena es texto entre un par de comillas dobles. Al colocar la cadena entre paréntesis de la función printf(), se imprime el texto (sin las comillas) en la consola. Las **funciones** y cadenas son temas tratados en lecciones posteriores. No se preocupe si no comprende completamente los conceptos. Lo importante a tener en cuenta es que así es como creamos la salida en la consola.

¿Qué pasa con el **\n** al final de la cadena? ¡Buena pregunta! Esto se denomina secuencia de escape y se usa para agregar un carácter no visual dentro de una cadena.

En este caso, **\n** agrega una nueva línea al final de la cadena. Prueba lo que sucede cuando lo colocamos entre Hello y World!:

```
printf("Hello\nWorld!");
```

Otra secuencia de escape es `\t`. Esto es equivalente a la tecla de tabulación e insertará espacios dentro de una cadena.

Pruebe que sucede cambiando el siguiente código:

```
printf("Hello\tWorld!");
```

## Ejercicio!

Obtén el siguiente output por pantalla (no tener en cuenta los colores):

```
2 tazas de harina para todo uso
1 taza: mantequilla sin sal  (temperatura ambiente)
2/3 tazas: azúcar granulada
```