The image features a dark blue background with several overlapping geometric shapes. On the left side, there is a blue parallelogram and a light green parallelogram, both tilted at an angle. Below these, there are several dark grey and blue diagonal stripes that create a sense of depth and movement.

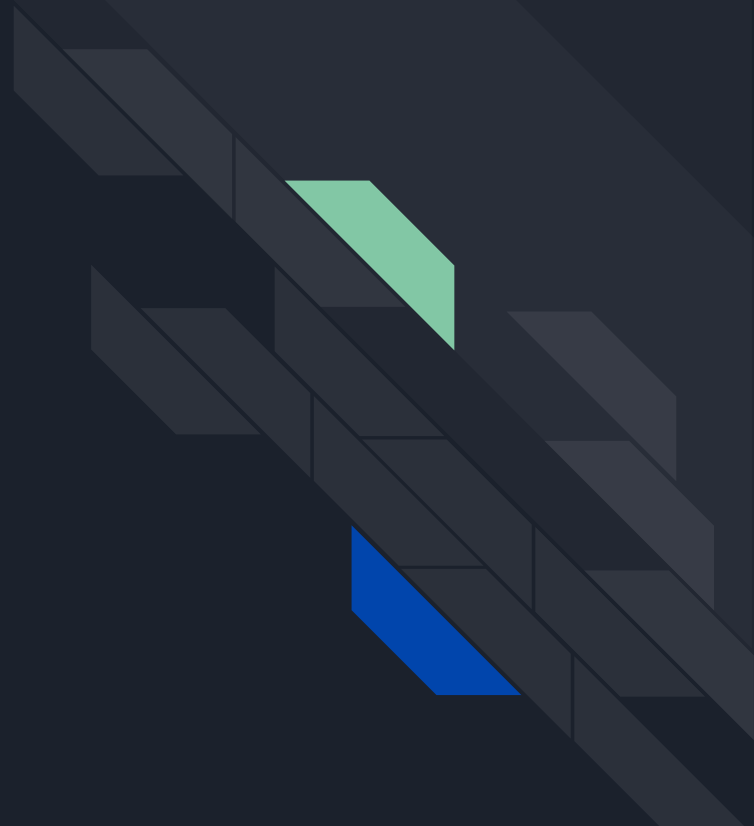
Algoritmos y Programación III

Mg. Claudio Aciti

claudioaciti@gmail.com

Ing. Martín Joel Rodríguez

mjrodriguez@untref.edu.ar



A decorative graphic on the left side of the slide. It consists of a blue parallelogram and a light green parallelogram, both tilted at an angle. The blue shape is in the foreground, and the green shape is partially behind it. They are set against a dark blue background with faint, lighter blue diagonal stripes.

Introducción al lenguaje C



Algunas características de C

- Compilado
- Bajo nivel (al menos más bajo nivel que Java)
- Tipo estructurado
- Tipado estático
- Acceso a memoria de bajo nivel mediante el uso de punteros
- Tipos de datos agregados (struct) que permiten que datos relacionados (como un empleado, que tiene un id, un nombre y un salario) se combinen y se manipulen como un todo (en una única variable "empleado")



C/C++ IDEs

- JetBrains CLion
- Code::Blocks
- Eclipse CDT(C/C++ Development Tooling)
- Netbeans for C/C++ Development
- CodeLite IDE
- Atom Code Editor
- Sublime Text Editor
- Geany



Hello, world

```
#include <stdio.h>

int main() {
    // test
    printf("Hello, World!\n");
    return 0;
}
```



Pasos en la compilación de C

1. Preprocessor
2. Compiler
3. Assembler
4. Linker



Preprocessor

```
# 1 "test.c"
# 1 "<built-in>"
# 1 "<command-line>"
# 31 "<command-line>"
# 1 "/usr/include/stdc-predef.h" 1 3 4
# 32 "<command-line>" 2
# 1 "test.c"
```

```
int main() {

    printf("Hello, World!\n");
    return 0;
}
```




Compiler

```
.file "test.c"
.text
.section      .rodata
.LC0:
.string      "Hello, World!"
.text
.globl main
.type main, @function
main:
.LFB0:
.cfi_startproc
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
leaq .LC0(%rip), %rdi
call puts@PLT
```

```
movl $0, %eax
popq %rbp
.cfi_def_cfa 7, 8
ret
.cfi_endproc
.LFE0:
.size main, .-main
.ident "GCC: (Ubuntu
7.3.0-27ubuntu1~18.04) 7.3.0"
.section
.note.GNU-stack,"",@progbits
```



Assembler

ELF>?@

UH??H?=??]Hello, World!GCC: (Ubuntu 7.3.0-27ubuntu1~18.04) 7.3.0zRx
R

?A?C

#test.cmain_GLOBAL_OFFSET_TABLE_puts????????

????????

.symtab.strtab.shstrtab.rela.text.data.bss.rodata.comment.note.GNU-stack.re

la.eh_frame @@0

&WW1W90e+B?W?R@@

?

?(Xa



Palabras reservadas

auto	else	long	switch
break	enum	register	typedef
case	extern	return	union
char	float	short	unsigned
const	for	signed	void
continue	goto	sizeof	volatile
default	if	static	while
do	int	struct	double



Tipos de datos

Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127
int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	4 bytes	-2,147,483,648 to 2,147,483,647
unsigned long	4 bytes	0 to 4,294,967,295



Tipos de datos

Type	Storage size	Value range	Precision
float	4 byte	$1.2\text{E}-38$ to $3.4\text{E}+38$	6 decimal places
double	8 byte	$2.3\text{E}-308$ to $1.7\text{E}+308$	15 decimal places
long double	10 byte	$3.4\text{E}-4932$ to $1.1\text{E}+4932$	19 decimal places



Formatos del printf

%c	character
%d	decimal (integer) number (base 10)
%i	integer (base 10)
%e	exponential floating-point number
%f	floating-point number
%o	octal number (base 8)
%s	a string of characters
%u	unsigned decimal (integer) number
%x	number in hexadecimal (base 16)



Caracteres especiales del printf

`\n` newline, or linefeed

`\r` carriage return

`\t` tab

`\v` vertical tab

`\a` audible alert

`\b` backspace

`\f` form feed

`\\` backslash



Entrada y salida

```
#include <stdio.h>
```

```
int main() {  
    char nombre[100];  
    printf("Ingrese su nombre: ");  
    scanf("%s", nombre);  
    printf("Su nombre es: %s", nombre);  
    return 0;  
}
```




Ejercicio 1: E/S

Programar una rutina que reciba nombre y edad de una persona y las imprima por pantalla.

Su nombre es: #####

Su edad es: ##



Ejercicio 1: E/S

```
#include <stdio.h>

int main() {
    char nombre[10];
    int edad;
    printf("Ingrese su nombre y edad: ");
    scanf("%s %d", nombre, &edad);
    printf("Su nombre es: %s \n", nombre);
    printf("Su edad es: %d \n", edad);
    return 0;
}
```



Ejercicio 2: Máximo

A partir de un listado de números, determinar el máximo.



Ejercicio 2: Máximo

```
#include <stdio.h>
```

```
int main() {  
    int numeros[] = {12, 2, 3, 4, 11, 6, 547, 8, 9};  
    int max = numeros[0];  
    for (int i = 1; i < sizeof(numeros) / sizeof(int); i++) {  
        if (numeros[i] > max) {  
            max = numeros[i];  
        }  
    }  
    printf("Máximo: %d", max);  
    return 0;  
}
```



Ejercicio 3: Mínimo

A partir de un listado de números, determinar el mínimo.



Ejercicio 3: Mínimo

```
#include <stdio.h>
```

```
int main() {  
    int numeros[] = {12, 2, -3, 4, 11, 6, 547, 8, 9};  
    int min = numeros[0];  
    for (int i = 1; i < sizeof(numeros) / sizeof(int); i++) {  
        if (numeros[i] < min) {  
            min = numeros[i];  
        }  
    }  
    printf("Mínimo: %d", min);  
    return 0;  
}
```



Ejercicio 4: Promedio

A partir del ingreso de tres enteros determinar el promedio.



Ejercicio 4: Promedio

```
#include <stdio.h>
```

```
int main() {  
    int ingreso[3];  
    int sum = 0;  
    printf("Ingrese 3 enteros: ");  
    scanf("%d %d %d", &ingreso[0], &ingreso[1], &ingreso[2]);  
    for (int i = 0; i < sizeof(ingreso) / sizeof(int); i++) {  
        sum = sum + ingreso[i];  
    }  
    float avg = (float) sum / sizeof(ingreso) * sizeof(int);  
    printf("Promedio: %f", avg);  
    return 0;  
}
```




Ejercicio 5: ASCII

Imprimir todos los caracteres ASCII.



Ejercicio 5: ASCII

```
#include <stdio.h>

int main() {
    for (int i = 0; i < 128; i++) {
        printf("%03d---%c \n", i, i);
    }
    return 0;
}
```



Ejercicio 6: Número par

Determinar si el número ingresado es par o no.



Ejercicio 6: Número par

```
#include <stdio.h>
```

```
int main() {  
    int numero;  
    printf("Ingrese un número: ");  
    scanf("%d", &numero);  
    if (numero % 2 == 0) {  
        printf("Es par.");  
    } else {  
        printf("No es par.");  
    }  
    return 0;  
}
```



Ejercicio 7: Menú

Diseñar un menú navegable donde cada opción muestre una frase distinta. Debe retornar al menú a menos que se elija la opción “Salir”.

1. Opción 1
2. Opción 2
3. Opción 3
4. Salir

Ingrese opción: #



Ejercicio 7: Menú

```
#include <stdio.h>
```

```
int main() {
```

```
    int opcion;
```

```
    do {
```

```
        printf("Menú \n");
```

```
        printf("1. Opción 1 \n");
```

```
        printf("2. Opción 2 \n");
```

```
        printf("3. Opción 3 \n");
```

```
        printf("4. Salir \n");
```

```
        scanf("%d", &opcion);
```

```
        switch (opcion) {
```

```
            case 1:
```

```
                printf("Ingresó a la opción 1 \n");
```

```
                break;
```

```
            case 2:
```

```
                printf("Ingresó a la opción 2 \n");
```

```
                break;
```

```
            case 3:
```

```
                printf("Ingresó a la opción 3 \n");
```

```
                break;
```

```
            case 4:
```

```
                break;
```

```
            default:
```

```
                printf("Opción no encontrada \n");
```

```
        }
```

```
    } while (opcion != 4);
```

```
    return 0;
```

```
}
```