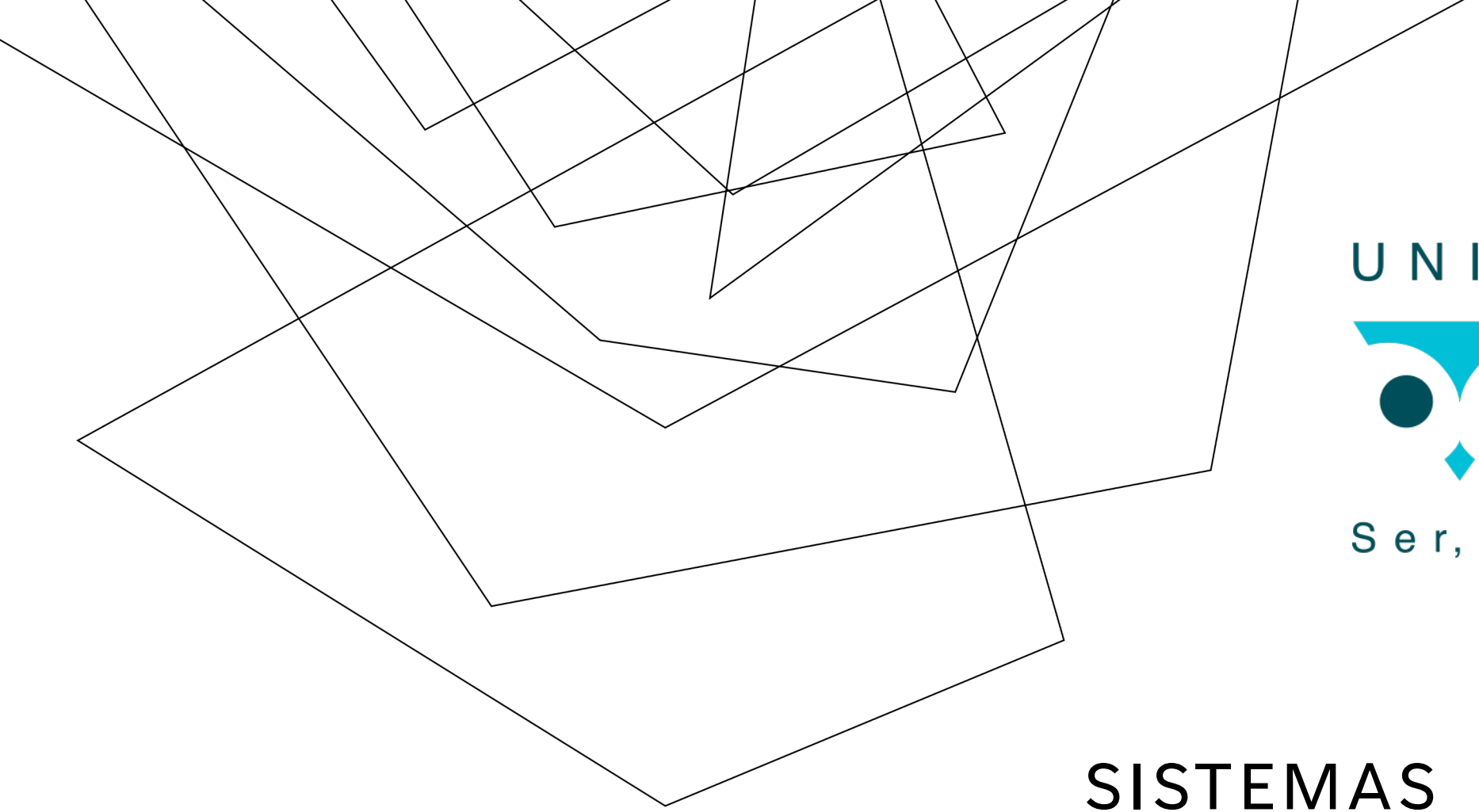

UNIVERSIDAD EIA®

Ser, Saber y Servir

SISTEMAS OPERATIVOS

Semestre 2022-2

INSTALACIÓN C WIN

https://sourceforge.net/projects/mingw/files/

https://code.visualstudio.com/

- Instalamos vscode
- Instalamos el compilador mingw
- Agregamos en las variables de entorno la ruta C:\MinGW\bin (o la ruta que usaste al instalar)
    - Variables de usuario editar Path y en este hacemos una nueva variable de entorno con la ruta anteriormente mencionada
- Abrir cmd y ejecutar gcc --version
- Abrimos vscode -> settings -> user -> extensions -> run code configuration -> ENABLE run in terminal
- Luego instalamos las extenciones de c/c++ de windows y la de code runner de jun han en vscode

UNIVERSIDAD EIA®

Ser, Saber y Servir

C ONLINE

https://replit.com/languages/c

UNIVERSIDAD EIA®

Ser, Saber y Servir

PARADIGMAS DE PROGRAMACIÓN

- Paradigma imperativo
  - Programación orientada a objetos
  - Programación procedimental
  - Programación estructurada
- Paradigma declarativo
  - Programación funcional
  - Programación lógica

UNIVERSIDAD EIA®

Ser, Saber y Servir

LENGUAJE C

- 1972 – Dennis Ritchie
- Lenguaje B
- Instrucciones tanto de nivel bajo como de nivel alto
- Lenguaje de Medio nivel, compilado (analiza, genera archivo binario y ejecuta), de propósito general, estructurado (ejecuta línea a línea) y modular, altamente portable, muy eficiente pero no permite funciones anidadas, problemas con la precedencia de operadores, nativamente no tiene programación multihilo (se necesitan librerías).

UNIVERSIDAD EIA®

Ser, Saber y Servir

LENGUAJE C

- Lenguaje tipado
- Apuntadores – son variables que contienen la dirección de una variable
- Usado en
  - Software de aplicación
  - Drivers
  - Sistemas operativos
  - Supercomputadoras
  - Sistemas embebidos
- https://www.youtube.com/watch?v=CYvJPra7Ebk&t=13s

UNIVERSIDAD EIA®

Ser, Saber y Servir

LENGUAJE C

- Que no encontraremos en C (nativo)
  - Hilos
  - Clases
  - Objetos
  - Recolección de basura
    - Funciones
      - malloc
      - free

UNIVERSIDAD EIA®

Ser, Saber y Servir

LENGUAJE C

- Comentarios
  - Una línea //
  - Varias líneas /* */
- Siempre cerramos con ;
- #Include – para invocar librerías
- <stdio.h> - librería base para funciones de entrada, salida y manipulación de ficheros
- Main – método base
- Una variable siempre debe empezar con una letra o _
- Se entiende que 0 es falso, cualquier otro valor es verdadero
- Statement – línea de código -> printf("Hola");
- Bloque de codigos – es un conjunto de statements que se agrupan por medio de {}
- Shift+Alt+F para formatear el codigo

UNIVERSIDAD EIA®

Ser, Saber y Servir

HOLA MUNDO

Vamos a crear una carpeta y en esta crear un archivo llamado hello.c

```c
#include <stdio.h>

int main()
{
    printf("hello");
    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

IMPRIMIR EN PANTALLA

- printf()

%d	int
%f	float
%lf	long float
%c	char
\n	Salto de línea
\t	tab

UNIVERSIDAD EIA®

Ser, Saber y Servir

TIPOS DE VARIABLES

Enteros	Int unsigned int short unsigned short long unsigned long	4 bytes 4 bytes 2 bytes 2 bytes 8 bytes 8 bytes	-2147483648 a 2147483647 0 a 4294967295 -32768 a 32767 O a 65535 -9223372036854775808 a 9223372036854775807 0 a 18446744073709551615
Punto flotante	float double long double	4 bytes 8 bytes 10 bytes	1,2e-38 a 3.4e+38 2,3e-308 a 1,7e+308 3,4e-4932 a 1,1e+4932
Caracteres	char unsigned char	1 byte 1 byte	-128 a 127 0 a 255
Vacio	void		

VARIABLES

- Declaración
  - int num;
- Asignación
  - int num= 1;

- Variables locales
  - Declaración dentro de la función main
- Variables globales
  - Declaración por fuera del main
- Variables externas
  - Declaración por fuera del main y con la palabra reservada extern al inicio de la declaración, igual también debemos declararlas en cada función que vayamos en la que vayamos a utilizarlas

UNIVERSIDAD EIA®

Ser, Saber y Servir

ENTEROS

```c
#include <stdio.h>

int main()
{
    int num1, num2;
    int num3 = 3;
    num1 = 2;
    printf("num3 es %d" , num3);
    printf ("num2 es %d y num1 es %d" ,
    num2, num1);
    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

PUNTO FLOTANTE

```c
#include <stdio.h>

int main()
{
    float pi = 3.1416;
    printf ("el valor de pi es %f", pi);
    printf ("el valor de pi es %.2f ", pi);
    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

CHAR

```
#include <stdio.h>

int main()
{
    char char1 = 'b';
    char char2 = 110;
    char char3[15] = “juan”;
    char char4[15];
    scanf(”%s ", &char4);
    printf("el valor del char1 es %c",
    char1);
    printf("el valor del char2 es %c",
    char2);
    printf("el valor del char3 es %s",
    char3);
    printf("el valor del char4 es %s",
    char4);

    return 0;
}
```

Tabla ASCII
https://upload.wikimedia.org/wikipedia/commons/1/1b/ASCII-Table-wide.svg

UNIVERSIDAD EIA®
Ser, Saber y Servir

CONSTANTES

entero	const int num = 223;
float	const float numf = 2.345;
char	const char charcons = ‘a’;

- #define num 8 -> constante simbólica – nunca se podrá cambiar

UNIVERSIDAD EIA®

Ser, Saber y Servir

OPERACIONES ARITMÉTICAS

+	Suma
-	Resta
*	Multiplicación
/	División - cociente
%	División – resto - modulo

UNIVERSIDAD EIA®

Ser, Saber y Servir

OPERACIONES LÓGICAS

Relación	
Mayor	>
Menor	<
Igual	==
Mayor que	>=
Menor que	<=
Distinto de	!=

Lógicos	
And	&&
Or	||
Not	!

UNIVERSIDAD EIA®

Ser, Saber y Servir

IF Y IF ANIDADO

```c
#include <stdio.h>

int main()
{
    int var = 3;

    if (var == 1)
        printf("el numero es 1");
    else if (var == 2)
        printf("el numero es 2");
    else if (var == 3)
        {
                printf ("el numero es 3 \n");
                printf("esto es un bloque");
        }
    else
        printf("no es ni 1 ni 2 ni 3");
    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

SWITCH Y SWITCH ANIDADO

```c
#include <stdio.h>

int main()
{
    int val = 3;

    switch (val)
    {
        case 0: printf("caso 1"); break;
        case 1: printf("caso 2"); break;
        default: printf("default"); break;
    }
    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

WHILE AND DO WHILE

```c
#include <stdio.h>

int main()
{
    int val = 3;

    while (val < 5)
    {
        val++;
        printf("entre en un while");
    }

    do{
        printf("entre al do");
    } while (val < 5);

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

FOR

```c
#include <stdio.h>

int main()
{
    for (int count = 1; count <= 5; count++)
    {
        printf("estoy en el for \n");
    }

    return 0;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

BREAK, CONTINUE, GOTO

```c
#include <stdio.h>

int main()
{
    int acum = 0;
    int count;
    for (count = 0; count <= 5; count++)
    {
        continue;
        printf("estoy en el for \n");
        acum = acum + 1;
    }

    printf("%d \n", count);
    printf("%d \n", acum);

    return 0;
}
```

Continue: cuando quieres seguir corriendo un ciclo pero por algún condicional quieres hacer skip de una iteración en específico

Goto: con esta instrucción controlamos errores

UNIVERSIDAD EIA®

Ser, Saber y Servir

BREAK, CONTINUE, GOTO

```c
#include <stdio.h>

int main()
{
    int i;
    for (i = 1; i <= 5; ++i)
    {
        printf("estoy en el for \n");
        if (i == 3)
        {
                goto jump;
        }
    }

    jump:
        printf("Salte");

    return 0;
}
```

Continue: cuando quieres seguir corriendo un ciclo pero por algún condicional quieres hacer skip de una iteración en específico

Goto: con esta instrucción controlamos errores

UNIVERSIDAD EIA®

Ser, Saber y Servir

FUNCIONES

```c
#include <stdio.h>

int func (int n);

int main()
{
    for (int count = 1; count <= 5; count++)
    {
        printf("estoy en el for y traigo de la funcion
        func el numero %d \n", func(count));
    }
    return 0;
}

int func (int n)
{
    printf("acabe de entrar a la funcion ");
    n++;
    return n;
}
```

UNIVERSIDAD EIA®

Ser, Saber y Servir

BIBLIOTECAS PRINCIPALES

```c
#include <stdio.h> // funcion i/o
#include <conio.h> // mejora rendimiento de i/o por consola
#include <string.h> // strings
#include <stdlib.h> // lib standard – permite comunicacion con el sistema
#include <math.h> // matematicas
#include <time.h> // fechas tiempo
#include <ctype.h> // manejo de caracteres
#include <signal.h> // señales (permite controlar una division por 0)
#include <locale.h> // configuraciones de entorno
#include <errno.h> // manejo de errores
#include <assert.h> // macro para deteccion de errores
#include <stdbool.h> // booleanos
#include <complex.h> // numeros complejos
#include <strcmp.h> // comparacion de cadenas
```

UNIVERSIDAD EIA®

Ser, Saber y Servir