



PROGRAMACIÓN I

TUP - 2024 - 2C - Turno noche - Presencial
Comisión 111

¿CUÁNDO Y DÓNDE?



Martes

18:00 a 22:00



Diagramación

Aula 201



Viernes

18:00 a 22:00



Codificación

Laboratorio 4

DOCENTES

- Maximiliano Sar Fernández - msarfernandez@docentes.frgp.utn.edu.ar
- José Alberto Rodríguez - jarodriguez@docentes.frgp.utn.edu.ar
- Javier Agustin Larroca - javier.larroca@alumnos.frgp.utn.edu.ar

Diagramación



DOCENTES

Codificación

- Javier Agustin Larroca - javier.larroca@alumnos.frgp.utn.edu.ar
- Jose Alberto Rodríguez - jarodriguez@docentes.frgp.utn.edu.ar
- Alejandro Martín Gómez Nieto - alejandro.gomez@alumnos.frgp.utn.edu.ar



CONTENIDOS

Etapla 1

1. Estructuras secuenciales

- Variables y constantes
- Operadores

2. Condicionales

- Simple
- Múltiple

3. Ciclos

- Ciclo exacto
- Ciclo inexacto

4. Ciclos combinados

Etapla 2

5. Funciones

- Diferentes tipos de parámetros

6. Vectores

- Carga y proceso
- Ordenamiento

7. Matrices y Struct

Proyecto final

CRITERIOS DE APROBACIÓN

Diagramación

- **Parcial 1** - Se aprueba con 6 o más, cuenta con su instancia de recuperación.
- **Parcial 2** - Se aprueba con 6 o más, cuenta con su instancia de recuperación.

Codificación

- **Tareas entregables** - Al finalizar cada guía de ejercicios se dará una tarea grupal a realizar en clase y entregar.
- **Trabajo Práctico Integrador** - Aplicación que deberán desarrollar de manera grupal y posteriormente defender. Tendrá un tiempo asignado para su desarrollo, resolver dudas y trabajar en el avance y una defensa. El TPI tiene una instancia de recuperación si su defensa no fuera satisfactoria.

HERRAMIENTAS PARA LA RESOLUCIÓN DE PROBLEMAS

Diagramación

Mediante diagramas de flujo, deberán ser capaces de representar la solución lógica a situaciones problemáticas.

Codificación

Mediante el lenguaje de programación C++, deberán ser capaces de desarrollar programas que ofrezcan soluciones a situaciones problemáticas.



SOFTWARES

Entorno y compiladores



CODEBLOCKS 20.03

WINDOWS Y LINUX

ZINJAI

WINDOWS

VISUAL STUDIO COMMUNITY

WINDOWS

DEV C++

WINDOWS

VISUAL STUDIO CODE + PLUGIN

WINDOWS, LINUX Y MAC

XCODE

MAC

¿CON CUÁLES TRABAJAREMOS?

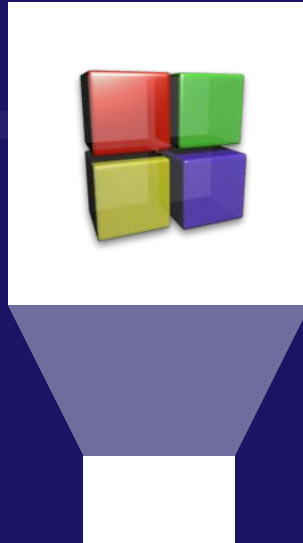
C++

Lenguaje de programación



Codeblocks

Entorno de trabajo y
compilador



Github

Repositorio y controlador
de versiones



CAMPUS VIRTUAL

- **Novedades**
- **Cafetería**
- **Foro de dudas generales**
- **Guía de cursada**
- **Equipo docente**
- **Repositorio**
- **Material de cada unidad**
- **Ejercicios de cada unidad**



moodle

<https://frgp.cvg.utn.edu.ar/>

- **Usuario:** DNI.frgp
- **Clave:** DNI

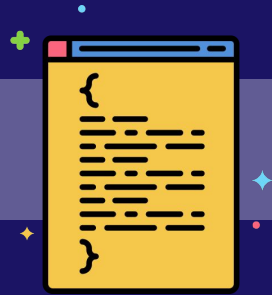


01

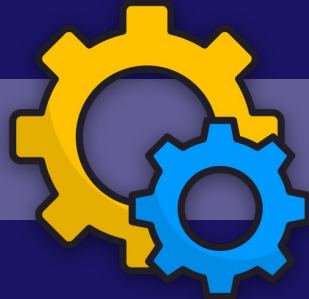
SECUENCIALES

DEL CÓDIGO AL PROGRAMA

- El proceso de transformar un **código fuente** en un **programa** que contiene las instrucciones que sean comprensibles por la computadora se llama **compilación**.



Código fuente



Compilación



Programa

CODIFICACIÓN

Cada elemento que utilicemos en un lenguaje de programación debe estar sujeto a una estricta sintaxis. Los elementos que el lenguaje admite son:

- **Variables y constantes**
- **Operadores**
- **Expresiones**
- **Palabras reservadas del lenguaje**

Como muchos lenguajes, C y C++ son case-sensitive. Esto significa que hace diferencias entre mayúsculas y minúsculas.

PALABRAS RESERVADAS DEL LENGUAJE

Palabras que el lenguaje utiliza para identificar tipos de datos, estructuras de programación, etc. Tienen un significado especial para el lenguaje y no pueden ser utilizados como identificadores de nombre.

Propósito	Palabras reservadas
Tipos de datos	bool, int, float, char, short, void, long, double
Elementos de programación	if, else, switch, default, break, for, while, do, return, auto, struct, class, static, virtual
Operadores	new, delete, sizeof

VARIABLES Y CONSTANTES

Representación simbólica de espacios de memoria. Es donde se almacenan los datos en procesamiento.

Una variable se identifica con un **tipo de dato** y un **identificador de nombre** (los elige el programador), y permite escribir un dato en la memoria o leer un dato de la memoria. Se puede modificar su valor las veces que sea necesario.

Una constante se identifica con la palabra reservada ***const***, un tipo, un nombre (lo elige el programador), y un valor que no puede ser modificado durante el transcurso del programa en ejecución.

VARIABLES Y CONSTANTES

Ejemplos de declaraciones de variables:

- `int` legajo;
- `float` sueldo;
- `string` nombre = "Programación I";
- `string` apellido;
- `char` letra;

Ejemplos de declaraciones de constantes:

- `const int` EDAD_MIN = 18;
- `const float` IMPUESTO = 10.5;
- `const char` PAIS = 'A';

OPERADORES MATEMÁTICOS

Necesarios para realizar cálculos matemáticos. Los paréntesis tienen el mismo efecto que en la matemática en las expresiones algebraicas. Sin embargo, en programación no se utilizan corchetes ni llaves para la separación de términos.

Operador	Operación
+	Suma
-	Resta
*	Producto
/	División real
%	Resto de la división entera

Ejemplos:

$$((2+3)*5)+10 \square 35$$

$$2+3*5+10 \square 27$$

$$5/2 \square 2$$

$$5.0/2 \square 2.5$$

$$5\%2 \square 1$$

OPERADORES DE ASIGNACIÓN

Nos permite asignar la expresión que se encuentra a la derecha del operador en la variable que se encuentra a la izquierda.

Formas correctas:

- edad = 50;
- a = b;
- nombre = "Jose";
- character = 'Y';
- precio = cant * pu;
- cont = 0;
- cont = cont + 1;
- cont = 0 + 1
- cont++;

Errores de sintaxis

- 50 = 50;
- 50 = edad;

Sintácticamente correcto pero sin sentido.

- edad = edad;

¿COMO EMPEZAR CON UN EJERCICIO?

Lo primero, antes de comenzar a resolver un ejercicio o problema, primeramente en diagrama y luego en código, es hacer un análisis del mismo en el que deberemos identificar:

1 Datos de entrada

2 Proceso

3 Datos de salida

EJEMPLO

1. Hacer un programa para ingresar por teclado la cantidad de horas trabajadas por un operario y el valor que se le paga por hora trabajada y listar por pantalla el sueldo que le corresponda.

1 Datos de entrada

- Horas trabajadas
- Valor por hora

2 Proceso

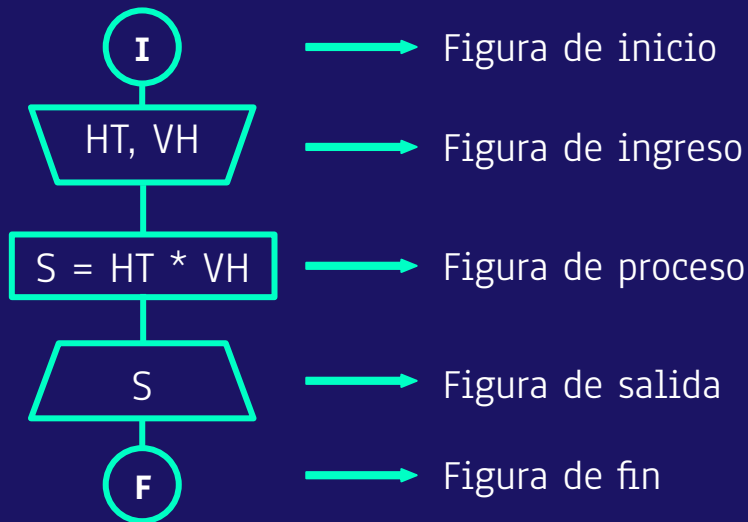
- Calcular el sueldo

3 Datos de salida

- Sueldo correspondiente

DIAGRAMACIÓN

1. Hacer un programa para ingresar por teclado la cantidad de horas trabajadas por un operario y el valor que se le paga por hora trabajada y listar por pantalla el sueldo que le corresponda.



DIAGRAMACIÓN

1. Hacer un programa para ingresar por teclado la cantidad de horas trabajadas por un operario y el valor que se le paga por hora trabajada y listar por pantalla el sueldo que le corresponda.

