



Técnicas de Programación I

TEMA 2

Paradigmas de la Programación

Profesor: José Luis Salazar

A G E N D A

1. Paradigmas de la programación.
2. Programación Secuencial.
3. Programación Estructurada.
4. Programación Modular.
5. Programación Recursiva.

1. Paradigmas de la programación.

Paradigmas de la Programación

Un paradigma de programación es un enfoque o estilo particular para escribir programas de computadora.

Cada paradigma tiene sus propias reglas, técnicas y filosofías que guían la forma en que se estructura y se escribe el código.



Paradigmas de la Programación

Tipos de Paradigmas

1. **Imperativa o procedimental:** Los programas consisten en una sucesión de instrucciones o conjunto de sentencias, como si el programador diera órdenes concretas. El desarrollador describe en el código paso por paso todo lo que hará su programa. Se centra en describir cómo realizar una tarea.



El primer lenguaje imperativo que permitió la creación de programas con un nivel de complejidad elevado fue FORTRAN. Hoy en día está representado por los lenguajes de programación BASIC, C ó PASCAL, entre otros.

Paradigmas de la Programación

Tipos de Paradigmas

2. **Funcional:** Utiliza funciones matemáticas puras evitando cambios y efectos secundarios.



El primer lenguaje de programación funcional fue LISP. Existen dos tipos de lenguajes funcionales: los puros (como HASKELL) y los híbridos (Scheme, Python, F#).

Paradigmas de la Programación

Tipos de Paradigmas

3. **Logica:** Se basa en la lógica matemática y la resolución de problemas a través de reglas y relaciones lógicas.

The logo for ProLog, featuring a stylized blue and black arrow pointing right, followed by the word "ProLog" in a bold, black, sans-serif font.

Se emplea en aplicaciones de inteligencia artificial. El lenguaje de programación lógica por excelencia es PROLOG, existen otros tales como: SQL y Erlang.

Paradigmas de la Programación

Tipos de Paradigmas

4. **Orientado a Objetos:** En este modelo de paradigma se construyen modelos de objetos que representan elementos (objetos) del problema a resolver, que tienen características y funciones. Las características del paradigma orientado a objetos son: encapsulamiento, abstracción, polimorfismo y herencia.

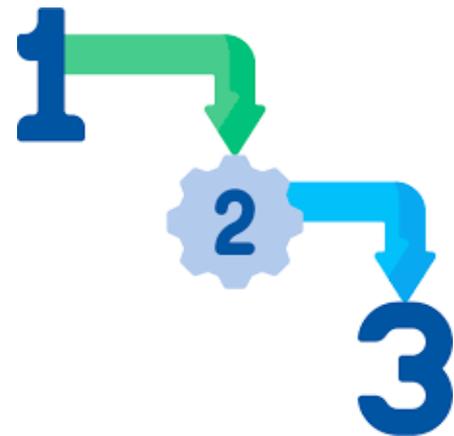


Algunos de los lenguajes de programación más populares que soportan la POO son: Java, Python, C++, C#, Ruby, PHP, JavaScript, y muchos otros.

2. Programación Secuencial

Programación Secuencial

La programación secuencial es un enfoque en el que las instrucciones se ejecutan una tras otra, en secuencia. Cada instrucción se ejecuta en el orden en que aparece en el código, sin saltos ni bifurcaciones. Este enfoque es común en la programación imperativa, donde las instrucciones se utilizan para modificar el estado del programa y realizar tareas paso a paso.



Programación Secuencial

Ejercicio

Calcular el factorial de un número entero positivo, con programación secuencial.

Programación Secuencial

Ejercicio resuelto

```
/*Autor(es): Jose Luis Salazar
Ejemplo factorial con programacion secuencial
Fecha de realización: {01-04-2025}
*/
#include <stdio.h>

int main() {
    int n;
    printf("Ingrese un número entero positivo: ");
    scanf("%d", &n);
    int factorial = 1;
    for (int i = 1; i <= n; i++) {
        factorial *= i;
    }
    printf("El factorial de %d es: %d\n",n,factorial);
    return 0;
}
```

3. Programación Estructurada

Programación Estructurada

La programación estructurada se basa en la idea de dividir un programa en estructuras lógicas más pequeñas y más manejables. Implica el uso de funciones para organizar el código y resolver problemas de manera eficiente.

En resumen, es una técnica que ayuda a construir programas más organizados, fáciles de entender y mantener, lo que a su vez puede reducir errores y mejorar la productividad de los desarrolladores.



Programación Estructurada

EJERCICIO

Calcular el factorial de un número entero positivo, con programación estructurada.

Programación Estructurada

EJERCICIO RESUELTO

```
/*Autor(es): Jose Luis Salazar
Ejemplo factorial con programacion estructurada
Fecha de realización: {01-04-2025}
*/
#include <stdio.h>

int calcularFactorial(int n);

int main() {
    int n;
    printf("Ingrese un número entero positivo: ");
    scanf("%d", &n);
    int resultado = calcularFactorial(n);
    printf("El factorial de %d es %d\n", n, resultado);
    return 0;
}

int calcularFactorial(int n) {
    int factorial = 1;
    for (int i = 1; i <= n; i++) {
        factorial *= i;
    }
    return factorial;
}
```

4. Programación modular.

Programación Modular

La programación modular es un enfoque que consiste en dividir un programa en módulos o unidades más pequeñas y autónomas. Cada módulo se encarga de una tarea específica y se comunica con otros módulos a través de interfaces bien definidas.



Programación Modular

Ejercicio

Calcular el factorial de un número entero positivo, con programación modular.

Programación Modular

Ejercicio Resuelto

```
/*Autor(es): Jose Luis Salazar
Ejemplo factorial con programacion modular
Fecha de realización: {01-04-2025}
*/
#include <stdio.h>
#include "factorial.h"

int calcularFactorial(int n); // Declaración de la
función

int main() {
    int n;
    printf("Ingrese un número entero positivo: ");
    scanf("%d", &n);
    int resultado = calcularFactorial(n);
    printf("El factorial de %d es %d\n", n, resultado);
    return 0;
}
```

Archivo main.c

Archivo factorial.h

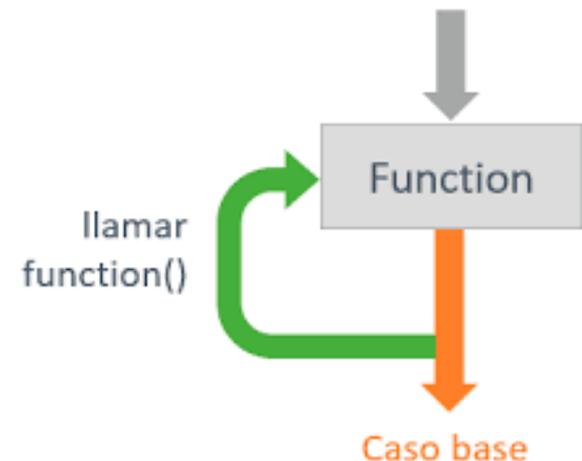
```
/*Autor(es): Jose Luis Salazar
Biblioteca factorial para programacion modular
Fecha de realización: {01-04-2025}
*/
#include <stdio.h>

int calcularFactorial(int n) {
    int factorial = 1;
    for (int i = 1; i <= n; i++) {
        factorial *= i;
    }
    return factorial;
}
```

5. Programación Recursiva

Programación Recursiva

En la programación recursiva, una función se divide en subproblemas más pequeños y se resuelve llamándose a sí misma con esos subproblemas. Cada llamada recursiva se acerca a la solución final del problema. La recursión se detiene cuando se alcanza una condición base o de salida que indica que ya no es necesario continuar llamándose a sí misma.



Programación Recursiva

Ejercicio

Calcular el factorial de un número entero positivo, con programación recursiva.

Programación Recursiva

Ejercicio resuelto

```
/*Autor(es): Jose Luis Salazar
Ejemplo factorial con programacion recursiva
Fecha de realización: {01-04-2025}
*/
#include <stdio.h>

int calcularFactorial(int n);

int main() {
    int n;
    printf("Ingrese un número entero positivo: ");
    scanf("%d", &n);
    int resultado = calcularFactorial(n);
    printf("El factorial de %d es %d\n", n, resultado);
    return 0;
}

int calcularFactorial(int n) {
    if (n == 0) {
        return 1; // Caso base
    } else {
        return n * calcularFactorial(n - 1); // Llamada recursiva
    }
}
```

5. Resumen

- ✓ **Programación Secuencial:** Código simple que ejecuta instrucciones en orden.
- ✓ **Programación Estructurada:** Uso de funciones para organizar el código.
- ✓ **Programación Modular:** División del código en archivos separados.
- ✓ **Programación Recursiva:** Uso de funciones que se llaman a sí mismas para resolver el problema.

Cada enfoque tiene sus ventajas y desventajas, y la elección del método depende del contexto y los requisitos del problema a resolver.

¿PREGUNTAS?

Ejercicios

Realizar los siguientes ejercicios, usando las técnicas dadas, de lectura, análisis, pseudocódigo y codificación en programación secuencial y estructurada.

1. Pedir al usuario un valor. Si el valor es positivo, pedir un segundo valor y calcular la suma, resta y producto de ambos. Mostrar los resultados por pantalla.
2. Un año es bisiesto si es divisible por 4 y no es por 100, o si es divisible por 400. Escribe un programa que lea un año y devuelva si es bisiesto o no.
3. Programa que lee las edades de un grupo de 100 alumnos y encuentra el promedio.

Ejercicios

Realizar los siguientes ejercicios, usando las técnicas dadas, de lectura, análisis, pseudocódigo y codificación en programación secuencial y estructurada.

4. Calcular de la suma y la media aritmética de N números reales. Solicitar el valor de N al usuario y cada uno de los N números reales.
5. La compañía que gestiona la autopista ha decidido cambiar las tarifas de peaje. Estas tarifas dependen del tipo de vehículo (camión o automóvil) que desee utilizar la autopista, del número de pasajeros que lleve y de la hora del día: en las horas de alta congestión los automóviles con tres pasajeros o más no pagan peaje, los camiones pagan 50 euros; en las horas menos congestionadas el peaje para los automóviles es de 25 euros y para los camiones es de 30 euros independientemente del número de pasajeros.
Escriba un programa que leyendo el tipo de vehículo, tipo de hora (si la hora es de alta congestión o no) y el número de pasajeros, calcule la tarifa que le debe cobrar a un vehículo.

Ejercicios

Realizar los siguientes ejercicios, usando las técnicas dadas, de lectura, análisis, pseudocódigo y codificación en programación secuencial y estructurada.

6. Calcular e imprimir la tabla de multiplicar de un número cualquiera. Imprimir el multiplicando, el multiplicador y el producto.
7. Al cerrar un expendio de naranjas, 15 clientes que aun no han pagado recibirán un 15% de descuento si compran más de 10 kilos. Determinar cuanto pagara cada cliente y cuanto percibirá la tienda por esas compras