Contenido

Ejemplos de usos IA

Codigos generados por la IA.

1. Programa en Java que toma dos números y devuelve la suma

```
java

import java.util.Scanner;

public class SumaNumeros {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Introduce el primer número: ");
        int num1 = sc.nextInt();
        System.out.print("Introduce el segundo número: ");
        int num2 = sc.nextInt();
        int suma = num1 + num2;
        System.out.println("La suma es: " + suma);
        sc.close();
    }
}
```

2. Página HTML en JSP para sumar dos números

suma.jsp

```
jsp

<%@ page contentType="text/html;charset=UTF-8" %>
<html>
<head><title>Suma JSP</title></head>
<body>
    <form method="get">
        Número 1: <input type="number" name="num1" required><br>
        Número 2: <input type="number" name="num2" required><br>
        <input type="submit" value="Sumar">
    </form>
    <%
        String n1 = request.getParameter("num1");
        String n2 = request.getParameter("num2");
        if (n1 != null && n2 != null) {
            int suma = Integer.parseInt(n1) + Integer.parseInt(n2);
            out.println("<p>Resultado: " + suma + "</p>");
        }
    %>
</body>
</html>
```

3. Página HTML + PHP para sumar dos números

suma.php

```
php

<!DOCTYPE html>
<html>
<head><title>Suma PHP</title></head>
<body>
    <form method="post">
        Número 1: <input type="number" name="num1" required><br>
        Número 2: <input type="number" name="num2" required><br>
        <input type="submit" value="Sumar">
    </form>
    <?php
        if ($_SERVER["REQUEST_METHOD"] == "POST") {
            $num1 = $_POST["num1"];
            $num2 = $_POST["num2"];
            echo "<p>Resultado: " . ($num1 + $num2) . "</p>";
        }
    ?>
</body>
</html>
```

4. Página HTML + JavaScript que suma dos números

```
html

<!DOCTYPE html>
<html>
<head><title>Suma JS</title></head>
<body>
    <input type="number" id="num1" placeholder="Número 1">
    <input type="number" id="num2" placeholder="Número 2">
    <button onclick="sumar()">Sumar</button>
    <p id="resultado"></p>

    <script>
        function sumar() {
            const a = parseFloat(document.getElementById('num1').value);
            const b = parseFloat(document.getElementById('num2').value);
            const resultado = a + b;
            document.getElementById('resultado').innerText = "Resultado: " +
resultado;
        }
    </script>
</body>
</html>
```

5. Página HTML en .NET (ASP.NET Razor Pages)

Suma.cshtml

```
html

@{
    Layout = null;
    int? resultado = null;
    if (IsPost) {
        int num1 = Convert.ToInt32(Request.Form["num1"]);
        int num2 = Convert.ToInt32(Request.Form["num2"]);
        resultado = num1 + num2;
    }
}
<!DOCTYPE html>
<html>
<head><title>Suma .NET</title></head>
<body>
    <form method="post">
        Número 1: <input type="number" name="num1" required><br>
        Número 2: <input type="number" name="num2" required><br>
        <input type="submit" value="Sumar">
    </form>
    @if (resultado.HasValue) {
        <p>Resultado: @resultado</p>
    }
</body>
</html>
```

6. Script SQL: Crear base de datos, tabla y usuario

```
sql

CREATE DATABASE ejemplo_db;
USE ejemplo_db;

CREATE TABLE personas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    edad INT
);

CREATE USER 'usuario_test'@'localhost' IDENTIFIED BY 'contrasena123';
GRANT ALL PRIVILEGES ON ejemplo_db.* TO 'usuario_test'@'localhost';
FLUSH PRIVILEGES;
```

7. Script SQL: Carga inicial en la tabla creada

```
sql

USE ejemplo_db;

INSERT INTO personas (nombre, edad) VALUES
('Ana', 25),
('Luis', 30),
('Marta', 22);
```

8. Script SQL: Eliminar usuario, tabla y base de datos

```
sql

DROP USER 'usuario_test'@'localhost';
DROP TABLE ejemplo_db.personas;
DROP DATABASE ejemplo_db;
```

9. CRUD java MVC

Resumen del CRUD MVC que haremos

Haremos una app para **gestionar personas** con estos datos:

- id (int)
- nombre (String)
- edad (int)

Podremos:

- Crear personas
- Leer (listar) personas
- Actualizar personas
- Eliminar personas

📂 **Estructura del proyecto (carpetas)**

css

CopiarEditar

CrudMVC/

├── controller/

│ └── PersonaController.java

├── model/

│ └── Persona.java

├── view/

│ └── PersonaView.java

└── Main.java

1. Modelo (model/Persona.java)

java

```
package model;

public class Persona {
  private int id;
  private String nombre;
  private int edad;

  public Persona(int id, String nombre, int edad) {
    this.id = id;
    this.nombre = nombre;
    this.edad = edad;
  }

  public int getId() { return id; }
  public String getNombre() { return nombre; }
  public int getEdad() { return edad; }

  public void setNombre(String nombre) { this.nombre = nombre; }
  public void setEdad(int edad) { this.edad = edad; }

  @Override
  public String toString() {
    return "ID: " + id + " | Nombre: " + nombre + " | Edad: " + edad;
  }
}
```

2. Vista (view/PersonaView.java)

java

```
package view;

import model.Persona;
import java.util.List;

public class PersonaView {

  public void mostrarPersona(Persona p) {
    System.out.println(p);
  }

  public void mostrarLista(List<Persona> personas) {
    for (Persona p : personas) {
      mostrarPersona(p);
    }
  }

  public void mostrarMensaje(String mensaje) {
    System.out.println(mensaje);
  }
}
```

3. Controlador (controller/PersonaController.java)

java

```
package controller;

import model.Persona;
import view.PersonaView;
import java.util.*;

public class PersonaController {
  private List<Persona> personas = new ArrayList<>();
  private PersonaView view = new PersonaView();

  public void crearPersona(int id, String nombre, int edad) {
    personas.add(new Persona(id, nombre, edad));
    view.mostrarMensaje("Persona creada.");
  }

  public void listarPersonas() {
    view.mostrarLista(personas);
  }

  public void actualizarPersona(int id, String nuevoNombre, int nuevaEdad) {
    for (Persona p : personas) {
      if (p.getId() == id) {
        p.setNombre(nuevoNombre);
        p.setEdad(nuevaEdad);
        view.mostrarMensaje("Persona actualizada.");
        return;
```

```
            }
        }
        view.mostrarMensaje("Persona no encontrada.");
    }

    public void eliminarPersona(int id) {
        personas.removeIf(p -> p.getId() == id);
        view.mostrarMensaje("Persona eliminada si existía.");
    }
}
```

4. Main (Main.java)

java

```
import controller.PersonaController;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        PersonaController controller = new PersonaController();
        Scanner sc = new Scanner(System.in);
        int opcion;

        do {
            System.out.println("\n=== CRUD Persona ===");
            System.out.println("1. Crear persona");
            System.out.println("2. Listar personas");
            System.out.println("3. Actualizar persona");
            System.out.println("4. Eliminar persona");
```

```
        System.out.println("0. Salir");
        System.out.print("Opción: ");
        opcion = sc.nextInt();
        sc.nextLine();

        switch (opcion) {
          case 1:
            System.out.print("ID: ");
            int id = sc.nextInt(); sc.nextLine();
            System.out.print("Nombre: ");
            String nombre = sc.nextLine();
            System.out.print("Edad: ");
            int edad = sc.nextInt();
            controller.crearPersona(id, nombre, edad);
            break;
          case 2:
            controller.listarPersonas();
            break;
          case 3:
            System.out.print("ID de la persona a actualizar: ");
            int idUpd = sc.nextInt(); sc.nextLine();
            System.out.print("Nuevo nombre: ");
            String nuevoNombre = sc.nextLine();
            System.out.print("Nueva edad: ");
            int nuevaEdad = sc.nextInt();
            controller.actualizarPersona(idUpd, nuevoNombre, nuevaEdad);
            break;
```

```
                case 4:
                    System.out.print("ID de la persona a eliminar: ");
                    int idDel = sc.nextInt();
                    controller.eliminarPersona(idDel);
                    break;
                case 0:
                    System.out.println("Saliendo...");
                    break;
                default:
                    System.out.println("Opción no válida.");
            }

        } while (opcion != 0);

        sc.close();
    }
}
```