

# CUADERNO PERSONAL

## DESARROLLO DE APLICACIONES WEB

### SEMANA 12 – Desarrollo Backend con Spring Core y Spring Boot



#### Desarrollo Backend

**Desarrollo Backend** se refiere a la parte del desarrollo de software que se encarga de la lógica del servidor, el manejo de datos y la comunicación con la base de datos. Es fundamental para crear aplicaciones web robustas y eficientes.

#### Arquitectura de Aplicaciones Web

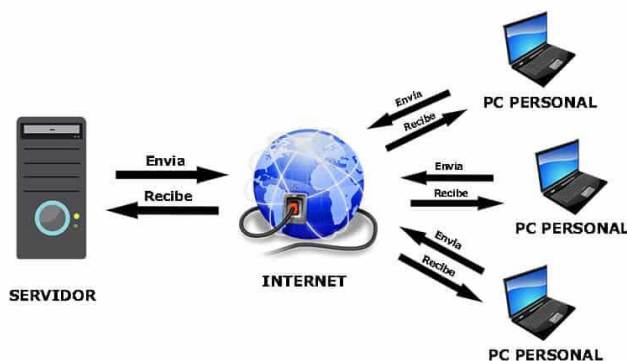
**Arquitectura de Aplicaciones Web** se refiere a la estructura y organización de los componentes que componen una aplicación web. Incluye:

##### 1. Servidor Web:

- Un servidor web es un software que maneja las solicitudes de los clientes y entrega páginas web y otros contenidos a través de HTTP/HTTPS. Ejemplos incluyen Apache HTTP Server y Nginx.

##### 2. Funcionamiento del Server Side:

- En el **server-side**, el servidor procesa las solicitudes del cliente, maneja la lógica de negocio, interactúa con la base de datos y genera las respuestas que se envían al cliente. Esto puede implicar la ejecución de scripts, el procesamiento de formularios y la generación de contenido dinámico.



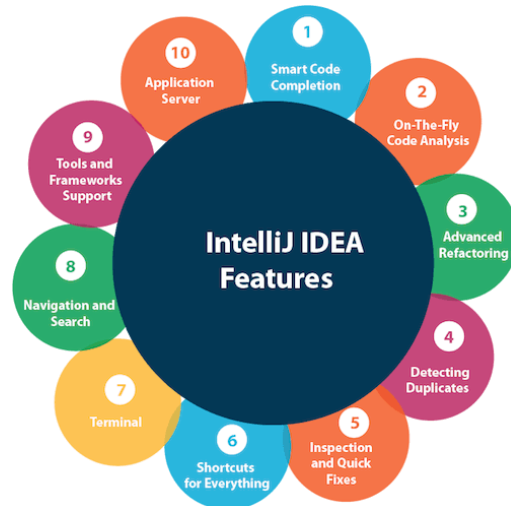
### 3. Lenguaje y Framework Backend:

- Los lenguajes de programación backend incluyen Java, Python, Ruby, PHP, entre otros. Los frameworks backend proporcionan herramientas y bibliotecas para simplificar el desarrollo. Ejemplos de frameworks incluyen Spring (Java), Django (Python) y Ruby on Rails (Ruby).

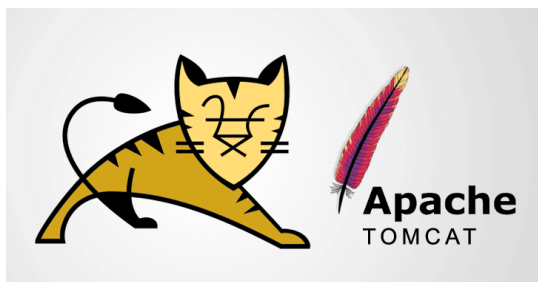
#### Java y Jakarta

**Java** es un lenguaje de programación ampliamente utilizado para el desarrollo backend, conocido por su portabilidad, robustez y escalabilidad.

**Jakarta EE** (anteriormente Java EE) es un conjunto de especificaciones que extiende la plataforma Java SE con especificaciones adicionales para el desarrollo de aplicaciones empresariales.



- **IntelliJ IDEA:**
  - **IntelliJ IDEA** es un entorno de desarrollo integrado (IDE) popular para Java y otros lenguajes. Ofrece características como completado de código, refactorización y soporte para herramientas de construcción y pruebas.
- **Tomcat:**
  - **Apache Tomcat** es un contenedor de servlets y un servidor web que implementa las especificaciones de Java Servlet y JavaServer Pages (JSP). Es utilizado para desplegar aplicaciones web basadas en Java.



#### Desarrollar Aplicaciones con Spring Core y Spring Boot

**Spring Framework** es un framework para el desarrollo de aplicaciones en Java que proporciona soporte integral para

el desarrollo de aplicaciones empresariales. **Spring Boot** es una extensión

de Spring que simplifica el proceso de configuración y despliegue de aplicaciones Spring.

## Terminología y Características de Spring

### 1. Inyección de Dependencias (DI):

- La inyección de dependencias es un patrón de diseño que permite a los objetos recibir sus dependencias desde el exterior en lugar de crearlas internamente. Esto promueve una mayor modularidad y facilita la prueba de componentes.

### 2. Aspect-Oriented Programming (AOP):



- AOP permite la separación de preocupaciones transversales, como el manejo de transacciones y la seguridad, mediante la definición de aspectos que se aplican a los componentes de la aplicación.

### 3. Contenedor de Beans:

- El contenedor de beans en Spring es responsable de la creación, configuración y gestión del ciclo de vida de los objetos que componen una aplicación.

## Módulos Spring

### 1. Spring Core Container:

- Proporciona la funcionalidad de inyección de dependencias y el contenedor de beans.

### 2. Spring AOP:

- Soporta la programación orientada a aspectos, permitiendo definir y aplicar aspectos a componentes.

### 3. Spring Data Access/Integration:

- Facilita el acceso a datos y la integración con diferentes fuentes de datos.

#### 4. **Spring Web:**

- Incluye soporte para el desarrollo de aplicaciones web, incluyendo la integración con tecnologías como servlets y JSP.

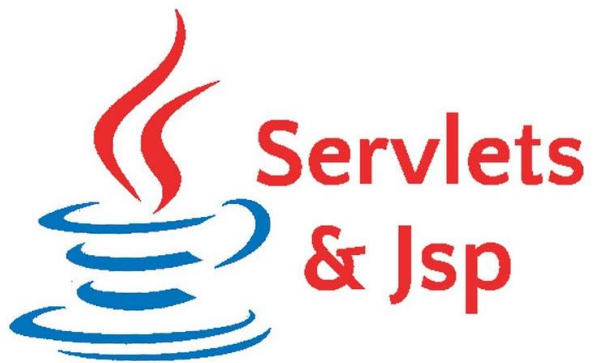
### **Servlets y JSP**

#### 1. **Servlets:**

- Los servlets son componentes Java que responden a las solicitudes de los clientes en una aplicación web. Se ejecutan en el servidor y generan contenido dinámico.

#### 2. **JSP (JavaServer Pages):**

- JSP es una tecnología que permite crear contenido web dinámico en Java. Las páginas JSP se convierten en servlets en tiempo de ejecución, permitiendo mezclar código Java con HTML.



### **Spring Boot**

**Spring Boot** es una herramienta que simplifica el proceso de configuración y despliegue de aplicaciones Spring. Proporciona una configuración automática y un conjunto de convenciones predeterminadas que facilitan la creación de aplicaciones listas para producción.

### **Características de Spring Boot**

#### 1. **Configuración Automática:**

- Spring Boot configura automáticamente los componentes necesarios basándose en las dependencias presentes en el proyecto.

## 2. Arranque Rápido:

- Permite iniciar aplicaciones rápidamente con configuraciones predeterminadas y un servidor embebido (como Tomcat o Jetty).

## 3. Dependencias Simplificadas:

- Utiliza un archivo pom.xml (Maven) o build.gradle (Gradle) simplificado para gestionar dependencias y configuraciones.

## 4. Spring Boot Starter:

- Proporciona conjuntos de dependencias y configuraciones comunes para diversos tipos de aplicaciones (por ejemplo, web, JPA, seguridad).

## 5. Ejemplo de Aplicación Spring Boot

### 6. Archivo de configuración principal (application.properties):

```
7. properties
8. Copy code
9. spring.datasource.url=jdbc:mysql://localhost:3306/mi_db
10. spring.datasource.username=usuario
11. spring.datasource.password=contraseña
12. server.port=8080
```

### 13. Clase principal de la aplicación:

```
14. java
15. Copy code
16. import org.springframework.boot.SpringApplication;
17. import
    org.springframework.boot.autoconfigure.SpringBootApplication;
18.
19. @SpringBootApplication
20. public class MiAplicacion {
21.     public static void main(String[] args) {
22.         SpringApplication.run(MiAplicacion.class, args);
23.     }
24. }
```

### 25. Controlador REST:

```
26. java
27. Copy code
28. import org.springframework.web.bind.annotation.GetMapping;
29. import org.springframework.web.bind.annotation.RequestMapping;
30. import org.springframework.web.bind.annotation.RestController;
31.
32. @RestController
33. @RequestMapping("/api")
34. public class MiControlador {
35.
36.     @GetMapping("/saludo")
37.     public String saludo() {
38.         return "Hola desde Spring Boot!";
39.     }
40. }
```

41. En este ejemplo, se configura una aplicación Spring Boot básica con una clase principal que arranca la aplicación y un controlador REST que expone un endpoint para recibir solicitudes HTTP.

Practica:

Código:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Datos Capturados</title>
  </head>
  <body>
    <h2>Datos capturados</h2>
    <p><strong>Nombres:</strong> <%= request.getParameter("nombre") %></p>
    <p><strong>Apellidos:</strong> <%= request.getParameter("apellidos")
%></p>
    <p><strong>Genero:</strong> <%= request.getParameter("genero") %></p>
    <p>
      <strong>Pasatiempos:</strong>
      <% String[] pasatiempos = request.getParameterValues("pasatiempos");
if
      (pasatiempos != null) { for (String pasatiempo : pasatiempos){
        out.print(pasatiempo + " - "); } } else { out.print("No
seleccionado"); }
      %>
    </p>
  </body>
</html>

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Formulario de Registro</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>

    <style>
      body {
        font-family: Arial, sans-serif;
        background-color: #121212;
        color: #ffffff;
        margin: 0;
        padding: 0;
        display: flex;
```

```
        justify-content: center;
        align-items: center;
        height: 100vh;
    }
    .container {
        background-color: #1e1e1e;
        padding: 20px;
        border-radius: 8px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);
        max-width: 400px;
        width: 100%;
    }
    h2 {
        text-align: center;
        color: #ffffff;
        margin-bottom: 20px;
    }
    label {
        display: block;
        margin-bottom: 8px;
        color: #bbbbbb;
    }
    input[type="text"],
    input[type="radio"],
    input[type="checkbox"] {
        margin-bottom: 15px;
    }
    input[type="text"] {
        width: 100%;
        padding: 8px;
        border: 1px solid #333333;
        border-radius: 4px;
        background-color: #2b2b2b;
        color: #ffffff;
    }
    input[type="submit"] {
        width: 100%;
        padding: 10px;
        border: none;
        border-radius: 4px;
        background-color: #5cb85c;
        color: #ffffff;
        font-size: 16px;
        cursor: pointer;
        transition: background-color 0.3s;
    }
    input[type="submit"]:hover {
        background-color: #4cae4c;
    }
}
```

```

        .inline {
            display: inline-block;
            margin-right: 10px;
        }
        input[type="radio"] + label,
        input[type="checkbox"] + label {
            margin-right: 10px;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Formulario de Registro</h2>
        <form action="guardar.jsp" method="get">
            <label for="nombre">Nombres:</label>
            <input type="text" id="nombre" name="nombre" required />
            <br />
            <br />
            <label for="apellidos">Apellidos:</label>
            <input type="text" id="apellidos" name="apellidos" required />
            <br />
            <br />
            <label>Genero:</label><br />
            <label for="masculino">Masculino</label>
            <input type="radio" id="masculino" name="genero" value="Masculino"
/>
            <label for="femenino">Femenino</label>
            <input type="radio" id="femenino" name="genero" value="Femenino"
/>
            <br />
            <br />
            <label>Pasatiempos:</label><br />
            <label for="futbol">Futbol</label>
            <input type="checkbox" id="futbol" name="pasatiempos"
value="Futbol" />
            <label for="natacion">Natacion</label>
            <input
                type="checkbox"
                id="natacion"
                name="pasatiempos"
                value="Natacion"
            />
            <label for="lectura">Lectura</label>
            <input
                type="checkbox"
                id="lectura"
                name="pasatiempos"
                value="Lectura"
            /><br /><br />

```



```

        <input type="submit" value="ENVIAR" />
    </form>
</div>
</body>
</html>

<!DOCTYPE html>
<html>
    <head>
        <title>Datos Capturados (POST)</title>
    </head>
    <body>
        <h2>Datos Capturados</h2>
        <p><strong>Nombre:</strong> <%= request.getParameter("nombre") %></p>
        <p><strong>Apellidos:</strong> <%= request.getParameter("apellidos")
%></p>
        <p><strong>Genero:</strong> <%= request.getParameter("genero") %></p>
        <p>
            <strong>Pasatiempos:</strong>
            <% String[] pasatiempos = request.getParameterValues("pasatiempos");
if
            (pasatiempos != null) { for (String pasatiempo : pasatiempos) {
            out.print(pasatiempo + " "); } } else { out.print("No seleccionados");
}
            %>
        </p>
    </body>
</html>

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8" />
        <title>Formulario de Registro</title>
        <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>

    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #121212;
            color: #ffffff;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }

```

```
.container {
  background-color: #1e1e1e;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);
  max-width: 400px;
  width: 100%;
}
h2 {
  text-align: center;
  color: #ffffff;
  margin-bottom: 20px;
}
label {
  display: block;
  margin-bottom: 8px;
  color: #bbbbbb;
}
input[type="text"],
input[type="radio"],
input[type="checkbox"] {
  margin-bottom: 15px;
}
input[type="text"] {
  width: 100%;
  padding: 8px;
  border: 1px solid #333333;
  border-radius: 4px;
  background-color: #2b2b2b;
  color: #ffffff;
}
input[type="submit"] {
  width: 100%;
  padding: 10px;
  border: none;
  border-radius: 4px;
  background-color: #5cb85c;
  color: #ffffff;
  font-size: 16px;
  cursor: pointer;
  transition: background-color 0.3s;
}
input[type="submit"]:hover {
  background-color: #4cae4c;
}
.inline {
  display: inline-block;
  margin-right: 10px;
}
```

```

        input[type="radio"] + label,
        input[type="checkbox"] + label {
            margin-right: 10px;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Formulario de Registro</h2>
        <form action="guardarPost.jsp" method="post">
            <label for="nombre">Nombres:</label>
            <input type="text" id="nombre" name="nombre" required />
            <br />
            <label for="apellidos">Apellidos:</label>
            <input type="text" id="apellidos" name="apellidos" required />
            <br />
            <label>Genero:</label><br />
            <label for="masculino">Masculino</label>
            <input type="radio" id="masculino" name="genero" value="Masculino"
/>
            <label for="femenino">Femenino</label>
            <input type="radio" id="femenino" name="genero" value="Femenino"
/>
            <br />
            <label>Pasatiempos:</label><br />
            <label for="futbol">Futbol</label>
            <input type="checkbox" id="futbol" name="pasatiempos"
value="Futbol" />
            <label for="natacion">Natacion</label>
            <input
                type="checkbox"
                id="natacion"
                name="pasatiempos"
                value="Natacion"
            />
            <label for="lectura">Lectura</label>
            <input
                type="checkbox"
                id="lectura"
                name="pasatiempos"
                value="Lectura"
            /><br /><br />
            <input type="submit" value="ENVIAR" />
        </form>
    </div>
</body>

```

```
</html>

<!DOCTYPE html>
<html>
  <head>
    <title>Mostrar Estudiantes</title>
    <style>
      body {
        font-family: Arial, sans-serif;
        background-color: #f4f4f9;
        margin: 0;
        padding: 0;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
      }
      .container {
        background-color: #fff;
        padding: 20px;
        border-radius: 8px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        text-align: center;
      }
      input[type="submit"] {
        padding: 10px 20px;
        border: none;
        border-radius: 4px;
        background-color: #5cb85c;
        color: #fff;
        font-size: 16px;
        cursor: pointer;
      }
      input[type="submit"]:hover {
        background-color: #4cae4c;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <h2>Mostrar Datos de Estudiantes</h2>
      <form action="mostrarEstudiantes.jsp" method="post">
        <input type="submit" value="Mostrar Estudiantes" />
      </form>
    </div>
  </body>
</html>

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
```

```

<!DOCTYPE html>
<html>
  <head>
    <title>Datos de Estudiantes</title>
    <style>
      body {
        font-family: Arial, sans-serif;
        background-color: #f4f4f9;
        margin: 0;
        padding: 0;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
      }
      table {
        width: 80%;
        margin: 20px auto;
        border-collapse: collapse;
      }
      table,
      th,
      td {
        border: 1px solid #ddd;
      }
      th,
      td {
        padding: 10px;
        text-align: left;
      }
      th {
        background-color: #f2f2f2;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <h2>Datos de Estudiantes</h2>
      <% String[][] estudiantes = { {"1", "Juan", "Pérez", "A"}, {"2",
"María",
  "García", "B"}, {"3", "Carlos", "López", "A"}, {"4", "Ana",
"Martínez",
  "C"}, {"5", "Pedro", "González", "B"}, {"6", "Sofía", "Rodríguez",
"A"},
  {"7", "Miguel", "Hernández", "C"}, {"8", "Lucía", "Fernández", "B"},
{"9",
  "Javier", "Ruiz", "A"}, {"10", "Isabel", "Torres", "C"} }; %>
      <table>
        <tr>

```

```

        <th>ID</th>
        <th>Nombre</th>
        <th>Apellido</th>
        <th>Grado</th>
    </tr>
    <% for (int i = 0; i < estudiantes.length; i++) { %>
    <tr>
        <td><%= estudiantes[i][0] %></td>
        <td><%= estudiantes[i][1] %></td>
        <td><%= estudiantes[i][2] %></td>
        <td><%= estudiantes[i][3] %></td>
    </tr>
    <% } %>
</table>
</div>
</body>
</html>

<!DOCTYPE html>
<html>
<head>
    <title>Agregar Estudiante</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <div class="container">
        <h1>Agregar Nuevo Estudiante</h1>
        <form action="insert.jsp" method="post">
            <label>Nombre:</label>
            <input type="text" name="nombre" required><br>
            <label>Apellido:</label>
            <input type="text" name="apellido" required><br>
            <label>Email:</label>
            <input type="email" name="email" required><br>
            <input type="submit" value="Agregar">
        </form>
        <a href="index.jsp">Regresar a la Lista</a>
    </div>
</body>
</html>

<%@ page import="java.sql.*" %>
<%
                                String                                url                                =
"jdbc:mysql://localhost:3306/escuela?useUnicode=true&characterEncoding=UTF-8"; // Cambia "escuela" por el nombre de tu base de datos
    String username = "root";
    String password = "admin";

```

```

        Connection conexion = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            conexion = DriverManager.getConnection(url, username, password);
        } catch (Exception e) {
            e.printStackTrace();
        }
    %>

<%@ include file="db.jsp" %>
<%
    int id = Integer.parseInt(request.getParameter("id"));
    String query = "DELETE FROM estudiante WHERE id=?";
    PreparedStatement pstmt = conexion.prepareStatement(query);
    pstmt.setInt(1, id);
    pstmt.executeUpdate();

    response.sendRedirect("index.jsp");
%>

<%@ include file="db.jsp" %>
<%
    int id = Integer.parseInt(request.getParameter("id"));
    String query = "SELECT * FROM estudiante WHERE id=?";
    PreparedStatement pstmt = conexion.prepareStatement(query);
    pstmt.setInt(1, id);
    ResultSet rs = pstmt.executeQuery();
    rs.next();
%>
<!DOCTYPE html>
<html>
<head>
    <title>Editar Estudiante</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <div class="container">
        <h1>Editar Estudiante</h1>
        <form action="update.jsp" method="post">
            <input type="hidden" name="id" value="<%= rs.getInt("id") %>">
            <label>Nombre:</label>
            <input type="text" name="nombre" value="<%=
rs.getString("nombre") %>" required><br>
            <label>Apellido:</label>
            <input type="text" name="apellido" value="<%=
rs.getString("apellido") %>" required><br>
            <label>Email:</label>
            <input type="email" name="email" value="<%=
rs.getString("email") %>" required><br>

```

```

        <input type="submit" value="Actualizar">
    </form>
    <a href="index.jsp">Regresar a la Lista</a>
</div>
</body>
</html>

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ include file="db.jsp" %>
<%
    // Consulta para obtener los datos de la tabla 'estudiante'
    String query = "SELECT * FROM estudiante";
    Statement stmt = conexion.createStatement();
    ResultSet rs = stmt.executeQuery(query);
%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Datos de la Tabla Estudiante</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <div class="container">
        <h1>Datos de la Tabla Estudiante</h1>
        <table>
            <tr>
                <th>ID</th>
                <th>Nombre</th>
                <th>Apellido</th>
                <th>Email</th>
                <th>Acciones</th>
            </tr>
            <%
                while (rs.next()) {
            %>
            <tr>
                <td><%= rs.getInt("id") %></td>
                <td><%= rs.getString("nombre") %></td>
                <td><%= rs.getString("apellido") %></td>
                <td><%= rs.getString("email") %></td>
                <td>
                    <a href="edit.jsp?id=<%= rs.getInt("id") %>">Editar</a>
                    <a href="delete.jsp?id=<%= rs.getInt("id")
%>">Eliminar</a>
                </td>
            </tr>
            <%
                }
            %>
        </table>
    </div>
</body>
</html>

```



```

        %>
        </table>
        <a href="add.jsp">Agregar Nuevo Estudiante</a>
    </div>
</body>
</html>
<%
    // Cerrar la conexión
    if (conexion != null) {
        try {
            conexion.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
%>

<%@ include file="db.jsp" %>
<%
    String nombre = request.getParameter("nombre");
    String apellido = request.getParameter("apellido");
    String email = request.getParameter("email");

    String query = "INSERT INTO estudiante (nombre, apellido, email) VALUES
(?, ?, ?)";
    PreparedStatement pstmt = conexion.prepareStatement(query);
    pstmt.setString(1, nombre);
    pstmt.setString(2, apellido);
    pstmt.setString(3, email);
    pstmt.executeUpdate();

    response.sendRedirect("index.jsp");
%>

```

Vista desde el LocalHost:

← → 📄 localhost:8080/semana12.1/ ☆

### Formulario de Registro

Nombres:

Apellidos:

Genero:  
☐ Masculino  
☐ Femenino

Pasatiempos:  
☐ Futbol  
☐ Natacion  
☐ Lectura

ENVIAR

← → 📄 localhost:8080/semana12.1/guardar.jsp?nombre=Cecilia&apellidos=Bracamonte&genero=Masculino&pasatiempos=Futbol&pasatiempos=Lectura

Datos capturados

Nombres: Cecilia  
Apellidos: Bracamonte  
Genero: Masculino  
Pasatiempos: Futbol - Lectura -

← → 📄 localhost:8080/semana12.2/ ☆

### Formulario de Registro

Nombres:

Apellidos:

Genero:  
☒ Masculino  
☐ Femenino

Pasatiempos:  
☐ Futbol  
☒ Natacion  
☐ Lectura

ENVIAR

## Datos Capturados

**Nombre:** Hugo

**Apellidos:** Chavez

**Genero:** Masculino

**Pasatiempos:** Natacion

### Mostrar Datos de Estudiantes

Mostrar Estudiantes

### Datos de Estudiantes

ID	Nombre	Apellido	Grado
1	Juan	Pérez	A
2	María	García	B
3	Carlos	López	A
4	Ana	Martínez	C
5	Pedro	González	B
6	Sofía	Rodríguez	A
7	Miguel	Hernández	C
8	Lucía	Fernández	B
9	Javier	Ruiz	A
10	Isabel	Torres	C

ID	Nombre	Apellido	Email	Acciones
1	Pedro	Biden	john.doe@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
2	Jane	Smith	jane.smith@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
3	Michael	Johnson	michael.johnson@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
4	Emily	Davis	emily.davis@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
5	David	Brown	david.brown@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
6	Sarah	Wilson	sarah.wilson@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
7	James	Taylor	james.taylor@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
8	Linda	Anderson	linda.anderson@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
9	Robert	Thomas	robert.thomas@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
10	Patricia	Jackson	patricia.jackson@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
12	Juanito	Almaña	almanaa@gmail.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
13	yanina	peres	yanina@gmail.com	<a href="#">Editar</a> <a href="#">Eliminar</a>

**Agregar Nuevo Estudiante**

Nombre

Apellido:

Email:

Agregar

**Agregar Nuevo Estudiante**

Nombre:

Juanito

Apellido:

Almaza

Email:

juanalmanin@gmail.com

Agregar

localhost:8080/semana12.5/index.jsp

Datos de la Tabla Estudiante

ID	Nombre	Apellido	Email	Acciones
1	Pedro	Biden	john.doe@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
2	Jane	Smith	jane.smith@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
3	Michael	Johnson	michael.johnson@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
4	Emily	Davis	emily.davis@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
5	David	Brown	david.brown@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
6	Sarah	Wilson	sarah.wilson@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
7	James	Taylor	james.taylor@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
8	Linda	Anderson	linda.anderson@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
9	Robert	Thomas	robert.thomas@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
10	Patricia	Jackson	patricia.jackson@example.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
12	Juanito	Alimaña	alimana@gmail.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
13	yanina	peres	yanina@gmail.com	<a href="#">Editar</a> <a href="#">Eliminar</a>
14	Juanito	Alimaña	juanalimanin@gmail.com	<a href="#">Editar</a> <a href="#">Eliminar</a>

[Agregar Nuevo Estudiante](#)

localhost:8080/semana12.5/edit.jsp?id=13

Editar Estudiante

Nombre

yanina

Apellido

peres

Email:

yanina@gmail.com

Actualizar

[Regresar a la Lista](#)