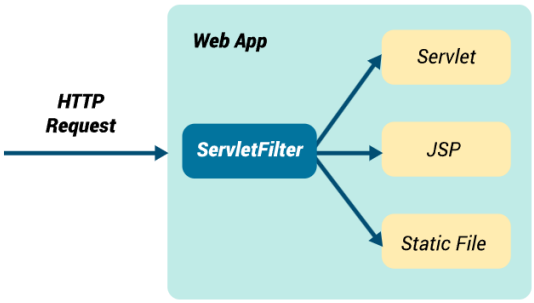
P4. Servlets y JSP

Los *Servlets* y los JSP (*Java Server Pages*), son tecnologías que se utilizan para realizar ejecuciones en servidores Java, permitiendo extender los servicios de un servidor *web*.

Un ejemplo sencillo de lo que se puede hacer con JSP y *Servlet* es que por medio de una página JSP se consigue hacer un formulario que contenga una serie de datos como: primer nombre, segundo nombre, edad, correo y, a través de un *servlet*, realizar el envío de estos registros para que sean almacenados en una base de datos y que, finalmente, el usuario pueda visualizar esta información.



P5. A continuación se pueden conocer los procesos principales de estas tecnologías y algunos aspectos generales del uso de JSP y *Servlet*:

***Los* servlets**

Un Servlet es un componente de Java que se ejecuta en el servidor y permite crear aplicaciones web dinámicas.

En palabras simples: es un programa en Java que corre dentro de un servidor web o de aplicaciones (como Apache Tomcat, GlassFish, WildFly, etc.) y que responde a las peticiones que hacen los clientes (normalmente navegadores) usando el protocolo HTTP.

Características principales:

- Forma parte de la especificación Java EE (Jakarta EE).
- Se ejecuta en el servidor (no en el navegador).
- Genera respuestas dinámicas (como HTML, JSON, XML, etc.).
- Maneja peticiones del cliente mediante los métodos GET, POST, PUT, DELETE de HTTP.
- Normalmente se usan junto con JSP o frameworks (Spring, JSF, Struts, etc.) para separar la lógica de negocio de la presentación.

Ciclo de vida de un Servlet:

1. Carga y creación → El contenedor web (ej. Tomcat) carga la clase del servlet y la instancia.
2. Inicialización (init()) → Se ejecuta una vez cuando arranca el servlet.
3. Ejecución (service()) → Atiende cada petición del cliente. Internamente llama a doGet(), doPost(), etc., según el tipo de solicitud.
4. Destrucción (destroy()) → Se llama cuando el contenedor lo elimina de la memoria.

JSP (*Java Server Pages*)

Es un código Java que se ejecuta en un servidor remoto, que lee las acciones del usuario desde formularios html, donde el servidor realiza consultas, usualmente a bases de datos, para obtener la información solicitada por el usuario y devolviendo una página *html* dinámica y en tiempo real.

es una tecnología de Java que permite crear páginas web dinámicas de forma más sencilla que usando solo Servlets.

👉 Se puede pensar como una combinación de HTML + Java, donde puedes escribir código Java embebido dentro de etiquetas especiales. El servidor (por ejemplo Apache Tomcat) convierte esa página JSP en un Servlet de manera automática.

Características principales de JSP:

- Está diseñado para generar contenido dinámico (HTML, XML, JSON, etc.).
- Usa Java en el lado del servidor y se traduce internamente en un Servlet.
- Permite mezclar código Java con etiquetas y HTML.
- Facilita separar la presentación (vista) de la lógica de negocio.
- Se puede integrar con JavaBeans, JDBC, frameworks web (Spring MVC, Struts, JSF, etc.).

Ciclo de vida de un JSP:

1. El cliente solicita una página JSP (index.jsp).
2. El servidor de aplicaciones traduce el JSP en un Servlet Java.
3. El servlet generado se compila y se ejecuta.
4. La salida (normalmente HTML) se envía al navegador del cliente.

Los archivos JSP

Están formados por código Java dentro de etiquetas **<html>** con extensión **.jsp** Estos archivos deben ser guardados en la carpeta del contenido *web* del servidor. El código JSP inicia con **<% %>.**

Los archivos JSP (JavaServer Pages) son archivos de texto con extensión .jsp que contienen una mezcla de:

- HTML o XML → para la estructura y presentación de la página.

- Código Java embebido → para generar contenido dinámico en el servidor.
- Etiquetas JSP y librerías (JSTL, custom tags) → para simplificar el uso de lógica sin escribir directamente código Java.

📌 Características de los archivos JSP:

1. Se almacenan en el servidor dentro de una aplicación web (ejemplo: /webapp/miarchivo.jsp).
2. Cuando un cliente solicita un .jsp, el contenedor web (Tomcat, GlassFish, etc.) lo traduce a un Servlet Java automáticamente.
3. El Servlet generado compila y ejecuta el código, y el resultado (normalmente HTML) es enviado al navegador.
4. Los archivos .jsp permiten separar la vista (interfaz de usuario) de la lógica de negocio.

P6. *Ejemplo*

Este es un ejemplo sencillo de código JSP.

```
<%@ page language='java' contentType='text/html;
charset=iso-8859-1'%>
<%@ page import='java.util.Date' %>
<html>
 <head>
  <title>Hola Planeta</title>
 </head>
 <body>
  <p>Bienvenido a la página JSP. </p>
  <p>La hora del servidor es <%= new Date() %> </p>
        <% out.println("HOLA PLANETA"); %>
 </body>
</html>
```

🔎 **Explicación línea por línea:**

1. **Directiva page**:

<%@ page language='java' contentType='text/html; charset=iso-8859-1'%>

- Indica que el lenguaje es **Java**.
- El contenido que se generará será **HTML**.
- Se especifica la codificación de caracteres (ISO-8859-1).

2. **Importar clases de Java**:

<%@ page import='java.util.Date' %>

- Importa la clase Date para poder mostrar la fecha y hora del servidor.

3. **HTML básico**:

```
<html>
<head>
  <title>Hola Planeta</title>
</head>
<body>
```

- Define la estructura de la página.

4. **Texto estático**:

```
<p>Bienvenido a la página JSP. </p>
```

- Texto fijo que siempre aparece.

5. **Expresión JSP (<%= ... %>)**:

```
<p>La hora del servidor es <%= new Date() %></p>
```

- Evalúa la expresión Java new Date() y la muestra directamente en el HTML.
- Ejemplo de salida:
  La hora del servidor es Thu Oct 02 11:50:10 COT 2025

6. **Scriptlet JSP (<% ... %>)**:

```
<% out.println("HOLA PLANETA"); %>
```

- Ejecuta código Java y escribe en la salida HTML.
- Mostrará en la página:
- HOLA PLANETA

📋 **Resultado en el navegador:**

Cuando se ejecute este JSP en un servidor como **Apache Tomcat**, verías algo parecido a:

Bienvenido a la página JSP.
La hora del servidor es Thu Oct 02 11:50:10 COT 2025
HOLA PLANETA

Software *requerido*

El *software* que se necesita para trabajar con JSP y *Servlets* es: **Java JEE IDE** (*NetBeans*, *Eclipse*), y el Servidor *Web* (*Tomcat*).

Otros requerimientos

Para crear ejecutar y probar *servlets* y páginas JSP, se debe configurar un entorno de desarrollo; las herramientas y pasos de configuración son: instalar el **JDK** (Kit de desarrollo De Java), instalar **IDE**
Netbeans en https://netbeans.apache.org/download/index.html, instalar Tomcat: Servidor de Aplicaciones, que es un *software* de código abierto; una vez terminada la configuración de las herramientas se inicia la introducción al tema de los *servlets*.

P7. ¿Qué es Apache Tomcat?

- Es un **servidor web y contenedor de servlets** desarrollado por la Fundación Apache.
- Sirve para ejecutar aplicaciones web desarrolladas en **Java** (Servlets, JSP y tecnologías relacionadas con Java EE).
- No es un servidor de aplicaciones completo como JBoss o GlassFish, pero es **ligero, rápido y muy usado** en proyectos web con Java.

En palabras simples: Tomcat es el programa que recibe las peticiones de los usuarios (por ejemplo desde un navegador) y ejecuta el código Java (servlets/JSP) que compone tu aplicación web.

🔷 ¿Qué se puede hacer con Tomcat?

1. **Desplegar aplicaciones web en Java** (.war o proyectos desde NetBeans/Eclipse/IntelliJ).
2. Manejar la comunicación entre el navegador y tu código Java.
3. Probar proyectos web en tu máquina local antes de subirlos a un servidor en producción.

http://localhost:8080

A continuación, se pueden conocer algunas ventajas de utilizar *servlet*:

- Rendimiento: a diferencia de CGI ***(Common Gateway Interface)*** el rendimiento de un *servlet* es mucho mejor.
- No necesita procesos independientes para operar las solicitudes del cliente.
- Ya que los *servlet* están escritos en Java los hace una plataforma independiente.
- Creíble: resguarda los recursos del servidor debido a el servidor de seguridad de Java.

P8. Las tareas principales de un *servlet* son:

Datos de transmisión

Se encarga de leer claramente los datos de transmisión de un cliente (navegador), incluyendo un formulario HTML.

Remisión de peticiones http

Remite una petición HTTP de los datos contenidos en el navegador incluyendo *cookies*.

Tratamiento de datos

Trata los datos y produce resultados accediendo a la base de datos.

Envío de documentos

Envía documentos al cliente (navegador), los cuales pueden ser de varios tipos como textos, binarios, etc.

Envíos de respuesta

Realiza envíos de respuesta HTTP implícita al cliente (navegador).

Ciclo de vida de un Servlet

A diferencia de las clases en Java, que inician con un método ***main ()***, en los *servlets* son "llamados" unos métodos, al recibir peticiones. A esta metodología se le llama ciclo de vida de un *servlet* y se compone por tres métodos: ***init, service, destroy*.**

P9. Formularios *HTML* con *Servlets*

Las páginas *web* contienen formularios *html* para la solicitud de información al usuario. El proceso de recolección de datos y respuesta, de este formulario, se debe hacer por medio de un programa. Esto lo hace un *servlet* que recoge las peticiones de un formulario y las procesa.

La declaración de un formulario HTML se hace por medio de la etiqueta **<form>,** donde es primordial que se especifique el método por el que se realizará el envío de los datos, los cuales pueden ser *get* o *post*.

A continuación, se presentan los aspectos más relevantes, en lo relativo a formularios *html* con *servlets*:

Ejercicio. Apache Tomcat y Formulario Web con JSP:

https://www.youtube.com/watch?v=_MLNWEd9GHQ

En este video se realizara un proyecto web, pero vamos a utilizar ahora el tomcat de Apache, lo primero que vamos a hacer es descargar Apache de tomcat en este link que tenemos acá del proyecto cuando ingresemos vamos a descargas la versión 9.0 37 y aquí

donde dice ZIP le van a dar ustedes descargar aquí y pues les va a empezar a descargar su

el servidor como tal comprimido, cuando termine de descargar este lo descomprimo y lo copiamos en el disco C: asi de sencillo.

ya lo tenía copia por eso no lo reemplazo. Ahora vamos a Netbeam ingresamos al menú de Tool – Server vamos a agregar un nuevo en el botón de abajo Add Server Interfaz, en la lista se escoje Apence Tomcat vamos siguiente aquí viene y nos pide donde tenemos la carpeta que acabamos de copiar en el disco C: Ahora nos pide un usuario y una contraseña yo le escribo admin – admin y ultimo finish.

aquí está la configuración del puerto, es mejor dejarlo como esta puerto 80 80.

Ahora vamos a crear un proyecto: File – New Proyect Java Web – web aplications siguientes le damos el nombre al proyecto Web_empresa_db siguiente y finalizar.

ya tenemos acá probemos sí está funcionando de forma adecuada cambiamos

aquí el html hola mundo venimos en el metro y luego cambiar el impar que lo podemos visualizar bien venimos a estar ejecutando y bueno y ahí está el mundo ahora vamos a quitar esto porque así como hacíamos vamos a utilizar una página jsp e vamos a eliminarlo

y aquí index.jsp vamos a colocar el index pero reconozca el apache tomcat como página de inicio venimos acá lo ejecutamos aquí está el agua y hasta el nivel cero que ahora lo que vamos a hacer es que vamos a aplicar un estilo a esta página le vamos a agregar booster buster le damos ahora vamos a utilizar vamos a darle iniciar de 234 4.5 el que vamos a utilizar le vamos a copiar y esto lo que queramos flaca y también aquí nos pide que cogemos a esto esto va adentro del body lo podemos poner acá no hay ningún problema y primeros poner un título acá es como la de un empleado y venimos y vamos a crear un día y le vamos a agregar un estilo de una vez clase igual donde sacamos esto como ya lo habíamos visto verdad tenemos la documentación oficial pues aquí nos va dando como que un un breve instructivo de todo lo que podemos hacer pero si no quieren algo como que más específico 3 describen 4 vamos acá y aquí están como que todos los ejemplos vienen que están por ejemplo desde contenedores y aquí nos dice verdad que utilizamos solo copiamos y pegamos y ahí está dentro de esta vamos a agregar un formulario vamos a agregar y es un fa estamos muy bien con él con el form vamos a agregar dos atributos que es el action que no vamos a utilizar todavía y el método que es post después de esto esto también lo vamos a agregar un class e igual va a ser de tipo font group clase yngling si nos

queremos como que buscar en en estos de acá verdad que nos podemos ir también a los de form y aquí también están como que los estilos en este caso vamos a utilizar el del axxion esto hacen como que un poquito diferente online pero que nosotros vamos a utilizar es el de esta forma igual si ustedes quieren elegir un estilo así como éste lo pueden lo pueden hacer a broma aquí sólo en forma clásica no estamos ya tenemos estos formularios de acabado cuando vamos a utilizar controles que pueden ser como que un input por ejemplo vamos a los 5 aquí están todos como que los tiempos que vienen tenemos aquí los el estilo y esto va agarrando los forma control entonces podemos utilizar este lady gaga copiar el estilo nuestra podemos poner cualquier nombre lvl en el primer campo que utiliza el código y aquí vamos a colocar códigos que hacer código esperado y podemos utilizar este camino si lo quieren es como que haciendo papito entonces vienen y colocan que es un input colocan el tipo de input que van a utilizar lo que en este caso es un mismo texto éste tiene dos atributos muy muy importantes que es el nombre que vamos a colocar txt código entonces esto es importantísimo abordar colocarle un nombre y como vamos a utilizar ya movilizar javascript y va a utilizar el el chico de vídeo de javascript entonces también le vamos a agregar lo que es una verdad entonces tenemos y le colocamos el mismo txt código después le agregamos así como es tener incluso un control y vivimos aquí claro un control después de esto ningún problema le vamos a colocar un place holder y que es requerido ahora extra place holder y esto crea como una pequeña marca de agua que sirve para era un ejemplo de cómo tener un ejemplo no si quieren corremos lo primero y después le agregamos este para que ustedes van viendo como como acá venimos a cambiar y aquí por ejemplo que código de esta forma entonces bien le agregamos olvido un place holder y colocamos ejemplo los puntos el código puede ser que es 001 damos guardar controles para ver los cambios y ya nos crea este marca de agua acá y cuando estés escribientes obviamente desaparecerla después vamos a indicar qué que este campo que este elemento de acá pues es requerido para que el momento que vemos clic en un botón pues bloque y ahora pues solo copiamos y pegamos y hacemos lo mismo con los demás controles venimos código aquí van a ser nombres y aquí va a ser nombres text nombres haydee text y nombres y acá vamos a colocar nombre nombre 2 y todo está bien medimos el estado y ahí vamos como que agregándole estilos nosotros quisiéramos como que resaltará más el esté acá le podemos agregar alegría a una gritar a la ética también esto ya es puro html básico no y seguimos vamos con apellidos y text lbl a pedir perdón y copiamos todo lo demás los pedidos apellidos y aquí vamos le ha pedido uno vi a pedrosa sólo con ejemplo tú i y este código nombres apellidos nos tocan dirección y dirección y tenemos aquí text dirección también junto con el ejemplo y sería número casa no causan calle para que los haremos vamos a tener una idea de que lo que va a llegar vamos con el de teléfono el nombre pedido dirección teléfono sólo vamos hacer un pequeño cambio en lugar de ser de tipo text vamos a decir que es de tipo duro y teléfono has teléfono y aquí sólo como ejemplo lo corremos y aquí la diferencia que esto es como de tipo número entonces sólo va a transmitir números no le va a permitir que ustedes escriben textos si copian y pegan textos y la tabla pero por lo menos escribirlo

y desde aquí para que vayan incrementando el número de 11 y ahora vamos con fechadas min en esta dirección y ahora vamos con fecha text ccn aquí también vienen en lugar de luna que vamos a cambiar a fecha y text fm y siempre explotando no vamos a colocar más arte para ustedes que no se va a ver y aquí me viene ya nos crea automáticamente un pequeño calendar bueno después de esto nosotros vamos a venir y vamos a al colocar un pequeño convoy aquí ya no va a ser un input sino que va a ser un se le igual las mismas características tiene un nombre le vamos a colocar rock un jardín su sangre igual es que el estilo y aquí pues le vamos a agregar ahorita por en todas las opciones le vamos a agregar manuales ya después lo vamos a agregar reconoce la base de datos el valor 1 opción 1 2 sí no 3 opción atrás vamos viendo cómo quedan vieron y ya tenemos a caja nos faltó agregar la etiqueta es anular la copiamos y estamos acá vamos a ver eleve el examen tipo sí y esta entonces acá bajito de esto vamos a agregar un botón verdad que es lo que nos va a servir en este barranco entonces si queremos saber cómo pueden utilizar por ejemplo un botón no vamos acá por la moto y aquí están como que todos los estilos bien vamos a utilizar este que dice primario y pues si quieren sólo copiamos y pega mis más fácil nos vamos acá y aquí va a ser agregar lo pueden hacer con un input también ahí es cuestión de gustos también le ponemos un nombre es importante que buscamos su nombre esto se llama de tn agregar igual y le colocamos el mismo dt n agregã en otros campos ya que ésta le vamos a agregar un pequeño salto de línea estamos aquí para tomar cualquiera de estos estilos o hacer una combinación de los mismos así como este el que más les les gusta si le quieren cambiar el tamaño pues sólo le le agregan esto aquí al final y si nos damos cuenta pues a ser un poquito más grande es cuestión de gustos de de caqui y en el otro vídeo vamos a llenar esto con con la base de datos muchas gracias

Codigo Index.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

  <head>

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <title>Empleados</title>

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-
TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
```

```
</head>
<body>
    <h1>Formulario Empleados</h1>
    <div class="container">
        <form action="" method="post" class="form-group" >
            <label for="lbl_codigo"><b>Codigo:</b></label>
            <input type="text" name="txt_codigo" id="txt_codigo" class="form-control" placeholder="Codigo: E001" required>
            <label for="lbl_nombres"><b>Nombres:</b></label>
            <input type="text" name="txt_nombres" id="txt_nombres" class="form-control" placeholder="Ejemplo: 1erNombre 2doNombre" required>
            <label for="lbl_apellidos"><b>Apellidos:</b></label>
            <input type="text" name="txt_apellidos" id="txt_apellidos" class="form-control" placeholder="Ejemplo: 1erApellido 2doApellido" required>
            <label for="lbl_direccion"><b>Direccion:</b></label>
            <input type="text" name="txt_direccion" id="txt_direccion" class="form-control" placeholder="Ejemplo: # Casa Barrio Ciudad" required>
            <label for="lbl_telefono"><b>Telefono:</b></label>
            <input type="number" name="txt_telefono" id="txt_telefono" class="form-control" placeholder="Ejemplo: 61 ### o 3###" required>
            <label for="lbl_fn"><b>Fecha Nacimiemto:</b></label>
            <input type="date" name="txt_fn" id="txt_fn" class="form-control" required>
            <label for="lbl_sangre"><b>Tipo de Sangre:</b></label>
            <select name="drop_sangre" id="drop_sangre" class="form-control">
                <option value="1">Opcion 1</option>
                <option value="2">Opcion 2</option>
                <option value="3">Opcion 3</option>
            </select>
            <br>
```

```
            <button type="button" name="btn_agregar" id="btn_agregar" class="btn btn-primary">Agregar</button>

        </form>


    </div>


    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-w1Q4orYjBQndcko6MimVbzY0tgp4pWB4lZ7lr30WKz0vr/aWKhXdBNmNb5D92v7s" crossorigin="anonymous"></script>

  </body>

</html>
```