

APLICACIONES WEB EN JAVA SERVER FACES

José Eduardo Rosas Ahuejote

Calle Francisco Mendoza Colonia Plan de Ayala Chilpancingo de los Bravo, Gro. 7472740944.
C.P. 39079 1140507@uagro.mx

M.C. Jorge Vázquez Galarce

Lázaro Cárdenas SN, Colonia Haciendita, Chilpancingo de los Bravo, Gro. 7471243942. C.P.
39079 13216@uagro.mx

M. en I. Rubén Rodríguez Camargo

Lázaro Cárdenas SN, Colonia Haciendita, Chilpancingo de los Bravo, Gro. 7471311223. C.P.
39079 ruben_rc@uagro.mx

M.C. León Julio Cortez Organista

Lázaro Cárdenas SN, Colonia Haciendita, Chilpancingo de los Bravo, Gro. 7471102752. C.P.
39079 ljcortez@uagro.mx

ABSTRACT

Web applications are everyday life in government, business, organizations, end users, children, education and all human endeavors, on the Internet. These applications use technologies that allow the remote interaction of actors, their processing and storage of large amounts of data. JavaServer Faces (JSF) is a framework for the development of multi-tier web applications, within the Java EE specification. These run server-side, are easy to build and maintain.

RESUMEN

Las aplicaciones web son el día a día en el gobierno, empresas, organizaciones, usuarios finales, niños, educación en todo el quehacer humano, en internet. Estas aplicaciones utilizan tecnologías que permiten la interacción a distancia de actores, su procesamiento y almacenamiento de grandes cantidades de datos. JavaServer Faces (JSF) es un framework para el desarrollo de aplicaciones de varios niveles web, dentro de la especificación Java EE. Estas se ejecutan del lado del servidor, son de fácil construcción y mantenimiento.

KEYWORDS

Java Server Faces, JavaBeans, JEE.

PALABRAS CLAVES

Java Server Faces, JavaBeans, JEE.

INTRODUCCIÓN

JavaServer Faces está orientado a la interfaz gráfica de usuario (GUI), facilitando el desarrollo de éstas, y realiza una separación entre el comportamiento y presentación de la aplicación. Esto lo hace a través de su propio servlet como controlador, implementando así los principios de diseño del Modelo-Vista-Controlador (MVC), lo que da como resultado un desarrollo más simple y una aplicación mejor estructurada.

JavaServer Faces está basado en un modelo de componentes y dirigido por eventos para el desarrollo de aplicaciones Web. Otra característica importante de JavaServer Faces es que, a pesar de que HTML es su lenguaje marcado por default, no está limitado a éste, pues tiene la capacidad de utilizar algún lenguaje diferente, para los componentes GUI y obtener así diferentes salidas para ser enviados al cliente.

El presente trabajo está conformado por tres capítulos: En el uno denominado Aplicaciones Web, se presenta el estado del arte, se explican conceptos claves, así como las tecnologías Web.

En el capítulo dos de título JavaServer Faces, se mencionan las plataformas y herramientas utilizadas, así como también los antecedentes, características, arquitectura y Funcionamiento.

En el tres, denominado caso de estudio se presenta la instalación de los componentes necesarios para el desarrollo de los ejemplos de JavaServer Faces, tales como el Kit de desarrollo de Java (JDK), el IDE Eclipse, el servidor de aplicaciones Apache Tomcat 8.5.

Finalmente, en las conclusiones, se presentarán los resultados principales de este trabajo de investigación.

1. APLICACIONES WEB

JavaServer Faces ofrece las bases para el desarrollo de aplicaciones de una sola página ofreciendo un contenido interactivo y de sencillo desarrollo a un costo razonable.

1.1 Conceptos clave.

WWW. Es el sistema de documentos de hipertexto que se encuentran enlazados entre sí y a los que se accede por medio de Internet. A través de un software conocido como navegador, los usuarios pueden visualizar diversos sitios web, los cuales contienen texto, imágenes, videos y otros contenidos multimedia y navegar a través de ellos mediante los hipervínculos. [1]

W3C. (World Wide Web Consortium) Es el comité que se dedica a implementar tecnologías uniformes en el uso y desarrollo de Internet. El organismo fue fundado en el MIT en Cambridge, Massachusetts, EE.UU. en 1994. [2]

HTML. Es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto. EL HTML se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto con diversos objetos (como fotografías, animaciones, etc.). [3]

XML. Significa eXtensible Markup Language, es decir, Lenguaje de Marcado Extendido, es un lenguaje de etiquetas que proviene del Lenguaje de Marcado Generalizado Estándar (SGML) y es regulado por el W3C. Son creadas por el programador, que estructuran y guardan de forma ordenada la información. No representa datos por sí mismo, solamente organiza la estructura.

Esto es que XML no es un lenguaje en sí mismo, sino un sistema que permite definir lenguajes de acuerdo a las necesidades. [4]

XHTML. De palabras inglesas (Extensible Hypertext Markup Language) que en español significa Lenguaje Extensible de Marcación Hipertexto, es un lenguaje muy similar al lenguaje HTML. Se puede adaptar y usar para muchos propósitos, sobre todo, la descripción de la apariencia de una página web. XHTML, es una aplicación particular de XML para “expresar” páginas web. [5]

1.2 Tecnologías para el desarrollo Web

Las tecnologías web avanzan rápidamente y es necesario estar al día de las tecnologías más

actuales. Estas permiten crear interfaces de usuario y establecer las comunicaciones con el servidor, además de implementar comportamientos de la web en el servidor.

CSS. Expresión inglesa Cascading Style Sheets, que se traduce como “Hojas de estilo en cascada”. El lenguaje CSS determina el estilo de los documentos HTML. Abarca opciones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, entre otros. Los lenguajes de hojas de estilo surgieron con la introducción de Internet y el crecimiento exponencial del lenguaje HTML para la creación de documentos electrónicos. [6]

JavaScript. Es una especie de lenguaje de programación ligera, interpretado por la mayoría de los navegadores y que les proporciona a las páginas web, efectos y funciones complementarias a las consideradas como estándar HTML. En un principio este lenguaje fue diseñado para añadir interactividad a las páginas webs y crear aplicaciones. A pesar de la similitud en el nombre, no está relacionado con Java. [7]

Java. Es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. Fue diseñado para ser multiplataforma y poder ser empleado el mismo programa en diversos sistemas operativos. Java es indispensable para el correcto funcionamiento de las diversas páginas web existentes en la actualidad, así como también para las aplicaciones de dispositivos Smartphone o computadoras. [8]

Java Server Pages. JSP, que en español significa Páginas de Servidor Java. Es una tecnología orientada a crear páginas web con programación en Java. Con JSP se pueden crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. [9]

PHP. Es el lenguaje que nos da la visualización de contenido dinámico en las páginas web. Todo el código PHP es invisible para el usuario, porque todas las interacciones que se desarrollan en este lenguaje son por completo transformadas para que se puedan ver imágenes, variedad de multimedia y los formatos con los que somos capaces de interactuar añadiendo o descargando información de ellos. [10]

ASP. (Active Server Pages) fue desarrollada por Microsoft para crear páginas dinámicas del servidor. ASP se escribe en la misma página web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft). [11]

1.3 Servidores Web

¿Qué es un servidor Web? Un servidor web es un software que forma parte del servidor y tiene como misión principal devolver información (páginas) cuando recibe peticiones por parte de los usuarios. Para el funcionamiento correcto de un servidor web necesitamos un cliente web que realice una petición http o https a través de un navegador como Chrome, Firefox o Safari y un servidor donde esté almacenada la información. [12]

¿Para qué sirve? Tras la primera consulta por parte del usuario hacia un sitio web, se establece una conexión entre el servidor DNS y la computadora que realiza la consulta o petición. Este servidor DNS responde con la dirección IP correcta del servidor web donde está alojado el contenido solicitado. El siguiente paso sería solicitar el contenido al servidor web mediante el protocolo HTTP/HTTPS. Una vez que el servidor web ha recibido la solicitud del contenido solicitado por el cliente web, deberá procesar la solicitud hasta encontrar el contenido solicitado dentro del dominio correspondiente. Envía el contenido solicitado al cliente web que lo solicitó. [12]

Tipos de servidores Web

Servidor NGINX. Este tipo de servidor web es conocido por su buen funcionamiento cuando tiene que gestionar un número alto de visitas simultáneas, ya que los usuarios no perciben retraso en la carga de la

página, aunque se esté produciendo un acceso concurrente. Actualmente es el software para servidores web más usado en el mundo. [12]

LITESPEED. LiteSpeed nació como sustituto de Apache, para mejorar el rendimiento del servidor web en entornos de alto tráfico. LiteSpeed ofrece total compatibilidad con Apache. [12]

Servidor Microsoft IIS. Este servidor web está desarrollado por Microsoft y es utilizado cuando se necesitan integrar propietarias de Microsoft. [12]

SUN JAVA System Web server. Este tipo de servidor está enfocado para programadores que trabajan con Java, Python o Ruby. Es un servidor web creado para soportar una gran carga de trabajo con tecnologías muy específicas como Java. [12]

APACHE. Apache es un servidor web de código abierto, multiplataforma y gratuito. Este web server es uno de los más utilizados en el mundo, actualmente el 43% de los sitios webs funcionan con él. Este servidor web desarrollado por Apache Software Foundation lleva en funcionamiento desde 1995. Una de las principales características de Apache es el uso del archivo .htaccess, utilizado entre todos los usuarios web. [13]

JOOMLA. Es un sistema de gestión de contenidos que puede ser utilizado independientemente. Entre sus principales virtudes está la de permitir integrar, añadir o editar el contenido de un sitio web de manera sencilla. Es de código abierto programado mayoritariamente en PHP bajo una licencia GPL. Este puede trabajar en redes locales o internet y requiere de una base de datos creada con un gestor MySQL, así como de un servidor HTTP Apache. [14]

1.4 Arquitectura cliente-servidor

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. [15]

Características:

Del lado del cliente:

- Envía peticiones.
- Espera las peticiones.
- Reciben contestaciones.

Del lado del servidor:

- Espera para las peticiones.
- Recibe las peticiones.
- Procesa las peticiones.
- Responde a los servicios solicitados. [16]

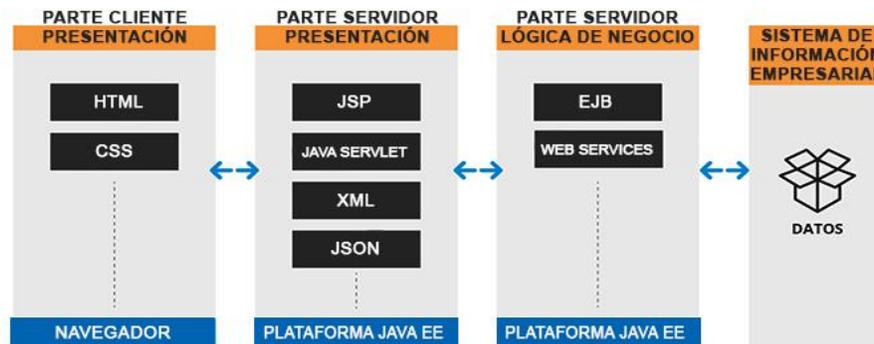
Ventajas:

- Facilidad de integración entre sistemas.
- Favorece el uso de interfaz de gráficas interactivas.
- La estructura modular facilita integración de nuevas tecnologías.
- Permite mantener un orden de trabajo, pero también poder acceder al mismo servidor e información que los demás.

Desventajas:

- Requiere habilidad para reparar un servidor.
- Protocolos de seguridad.
- Al ser computadoras de alto nivel generan limitaciones en costos económicos. [17]

Uso y aplicación. La arquitectura cliente-servidor se utiliza en sitios web que se encuentran en intranets e internet. El Cliente normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos. [18]



1.5 SERVIDOR DE APLICACIONES

TOMCAT. Este es un software que permite que un servidor web maneje contenido web dinámico basado en Java utilizando el protocolo HTTP. Se ha convertido sin duda en el estándar de facto para entornos de desarrollo web. Al reunir todas estas tecnologías basadas en Java, Tomcat ofrece un entorno de servidor web para ejecutar aplicaciones creadas en el lenguaje de programación Java. Funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Oracle Corporation. [19]

JBOSS. WildFly o anteriormente conocido como JBoss AS, o simplemente JBoss, es un servidor de aplicaciones Java EE de código abierto implementado con la especificación Java EE. En el año 2000 fue lanzada la primera versión JBoss AS, en el año 2013 el proyecto fue renombrado WildFly (WildFly 8), y así surgieron varias versiones hasta hoy. [20]

ORACLE WEBLOGIC. Oracle WebLogic Server es el servidor de aplicaciones líder de procesamiento de transacciones en línea de la industria para crear aplicaciones empresariales con los estándares de Java EE y desplegarlas con un tiempo de ejecución fiable y escalable y bajo costo de propiedad. [21]

2. JAVASERVER FACES.

2.1 JAVA Enterprise Edition. Es un conjunto de estándares de tecnologías dedicadas al desarrollo de Java del lado del servidor. La plataforma Java EE consta de un conjunto de servicios, API y protocolos que proporcionan la funcionalidad necesaria para desarrollar aplicaciones basadas en web de varios niveles. [22]

Figura 1. Esquema de la plataforma Java EE. [22]

2.2 Modelo vista controlador



Figura 2. Diagrama del Modelo-Vista-Controlador. [23]

Es un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista, o a cualquier parte del sistema puedan ser hechas con un mínimo impacto en el componente del modelo de datos o en los otros componentes del sistema. [24]

2.3 Capa de presentación o Web. Es una abstracción de la aplicación, y son los artefactos de código fuente en ejecución que verán los usuarios en otras palabras las vistas (las paginas JSP). Así mismo se puede llamar capa de presentación al conjunto de módulos que forman las vistas, controlador y la interfaz proporcionada por la capa de servicio (front-end). Y para el caso se tienen los servlets y la pagina JSP. [25]

2.3.1 Servlet. Los servlet es una tecnología que implementa el controlador de la aplicación respondiendo a peticiones HTTP. El servlet vive en el contenedor de servlets del lado del servidor de aplicaciones. [26]

2.3.2 Paginas JSP. Una página JSP es una página con sintaxis HTML que incorpora código Java en forma de una pieza de código de software que utiliza un lenguaje de scripting nativo de una página web (scriptlet) para generar el contenido dinámico. JSP (Java Server Pages). Para agregar el código Java dentro de la página JSP, se puede hacer de la siguiente manera:

-Expresión: “< % = ... % >”. Con dicho fragmento se le indica al intérprete de JSP que debe escribir en la salida lo que indique la expresión. Indicar que la expresión no debe acabar en punto y coma (;).

-Scriptlet: “<% ... % >” Dentro del scriptlet se puede incluir cualquier código Java.

-Declaraciones: <%! ... %> Sirve para declarar nuevas variables en la página JSP. [27]

2.4 JAVABEANS. Los Enterprise JavaBeans (EJB) son componentes JEE diseñados para encapsular, resolver y ejecutar la lógica y las reglas del negocio de la aplicación. Un bean es una clase Java escrita siguiendo unas ciertas convenciones. Estas convenciones hacen posible que herramientas automáticas puedan acceder a sus propiedades y manipularlas sin necesidad de modificar el código.

Se trata de una clase que contiene unas variables que van a almacenar la información que es necesaria. Éstas variables deben ser privadas. Y para acceder a éstas deben implementarse unos métodos get (para obtener el valor) y set (para establecer el valor). Éstos métodos deben ser públicos, además de un constructor. [28]

Los EJB son componentes pensados para desarrollar aplicaciones distribuidas, escalables, transaccionales y seguras. Esto hace que sea una aplicación Enterprise. La especificación define tres tipos de EJB: Session beans, Entity beans y Message-Driven beans.

2.5 ¿Qué es JAVA Server Faces? Es un framework, incluido dentro de la especificación Java EE, que tiene como misión facilitar la construcción y mantenimiento de aplicaciones Web en Java, siguiendo la arquitectura Modelo Vista Controlador. [29]

2.5.1 Antecedentes. JavaServer Faces es el framework oficial de Java Enterprise para el desarrollo de interfaces de usuario avanzadas en aplicaciones web. La especificación de JSF ha evolucionado desde su lanzamiento en 2004 y se consolidó, con nuevas características y funcionalidades. [30]

Tabla 1. Versiones JavaServer Faces. [31]

| | |
|-----------------------------|--|
| JSF 1.0 (11-03-2004) | Lanzamiento inicial de las especificaciones de JSF. |
| JSF 1.1 (27-05-2004) | Lanzamiento que solucionaba errores. Sin cambios en las especificaciones ni en el renderkit de HTML. |
| JSF 1.2 (11-05-2006) | Lanzamiento con mejoras y corrección de errores. |
| JSF 2.0 (12-08-2009) | Lanzamiento con mejoras de funcionalidad, rendimiento y facilidad de uso. |
| JSF 2.1 (22-10-2010) | Lanzamiento de mantenimiento, con mínimos cambios. |
| JSF 2.2 (16-04-2013) | Lanzamiento que introduce soporte a HTML 5, Faces Flow Stateless veas y Resource library contracts. |
| JSF 2.3 (28-03-2017) | Lanzamiento que introduce mayor soporte de CDI. Websockets, expresiones de búsqueda de componentes soporte básico de URLs sin extensiones y validación de Beans a nivel de clase. |

2.5.2 Características

JSF proporciona las siguientes características destacables:

- Definición de las interfaces de usuario mediante vistas que agrupan componentes gráficos.
- Conexión de los componentes gráficos con los datos de la aplicación mediante los denominados beans gestionados.
- Conversión de datos y validación automática de la entrada del usuario.
- Navegación entre vistas.
- Internacionalización.
- A partir de la especificación 2.0 un modelo estándar de comunicación AJAX entre la vista y el servidor. [32]

2.5.3 Arquitectura de JAVASERVER FACES. El navegador realiza una petición a una determinada URL en la que reside la página JSF que se quiere mostrar. En el servidor un servlet que llamamos motor de JSF recibe la petición y construye un árbol de componentes a partir de la página JSF que se solicita.

Este árbol de componentes replica en forma de objetos Java la estructura de la página JSF original y representa la estructura de la página que se va a devolver al navegador.

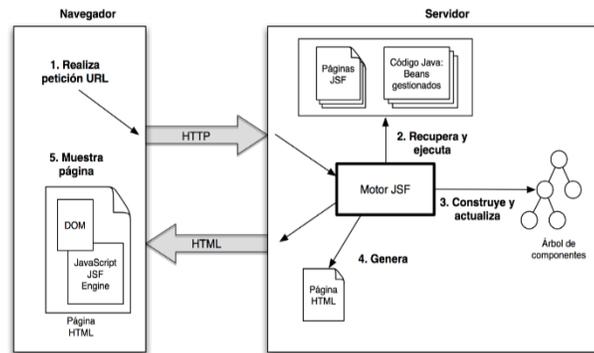


Figura 3. Funcionamiento de JSF para la generación de una página. [32]

Una vez que la página se muestra en el navegador, el usuario interactúa con ella. En este momento es cuando se utiliza el enfoque de AJAX. Los componentes contienen código JavaScript que se encarga de gestionar la interacción y pasar al servidor las peticiones específicas.

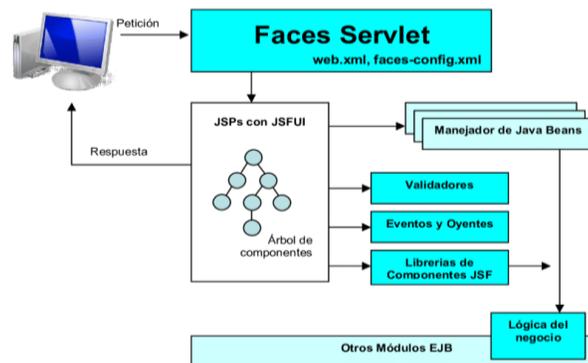


Figura 4. Proceso de JSF cuando la página ya está generada. [32].

3. CASO DE ESTUDIO.

3.1 Etiquetas JSF

Las vistas JSF se construyen mediante etiquetas específicas que declaran elementos y componentes. JSF brinda una biblioteca de tags o etiquetas HTML, que brindan un comportamiento especial y más potente que las etiquetas HTML estándar.

Para poder usarlas se debe agregar la siguiente línea dentro de la etiqueta html, allí vemos que se importa mediante el prefijo h para html.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
<html xmlns="http://www.w3.org/1000/xhtml"
xmlns:ui="http://java.sun.com/jsf/facelets"
xmlns:f="http://java.sun.com/js/faces"
xmlns:h="http://java.sun.com/jsf/html"

```

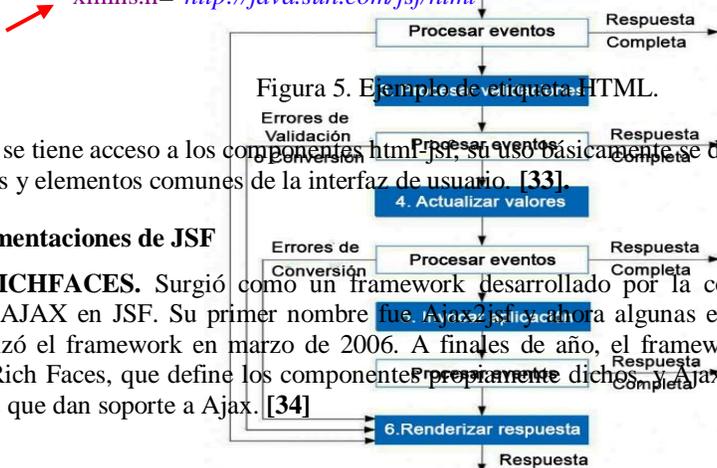


Figura 5. Ejemplo de etiquetas HTML.

Al hacerlo se tiene acceso a los componentes html-jsf, su uso básicamente se da para la construcción de formularios y elementos comunes de la interfaz de usuario. [33].

3.2 Implementaciones de JSF

RICHFACES. Surgió como un framework desarrollado por la compañía Exadel orientado a introducir AJAX en JSF. Su primer nombre fue Ajax.jsf, ahora algunas etiquetas tienen el prefijo a4j. Exadel lanzó el framework en marzo de 2006. A finales de año, el framework se dividió en dos partes llamados Rich Faces, que define los componentes de programación y Ajax4jsf, que define las etiquetas específicas que dan soporte a Ajax. [34]

ICEFACES Es un marco de desarrollo de aplicaciones de Internet enriquecidas (RIA) de código abierto para Java EE. Mejora la eficiencia del desarrollador al mismo tiempo que reduce el tiempo de comercialización y los costos operativos. Sus funciones y capacidades permiten a los desarrolladores hacer más de lo que se puede imaginar dentro de los límites de una infraestructura heredada. [35]

PRIMEFACES. Su primera versión fue lanzada el día 23 de febrero del año 2009 y es uno de los frameworks web con mayor aceptación hoy día y que muchos programadores web han puesto en práctica con resultados favorables. Primefaces es una librería de componentes visuales Open Source para JSF, así como IceFaces o RichFaces. Primefaces es el número dos de los diez frameworks más populares del mundo. [36]

3.3 Ciclo de desarrollo o de validación. El ciclo de vida es una secuencia de fases. Por las que pasa una petición JSF desde que se recibe en el servidor hasta que se genera la página HTML resultante. [37]

| Librería de etiquetas | Descripción | Ejemplo |
|-----------------------|--|--|
| JSTL Core | Etiquetas estándar JSP | <c:forEach> <c:catch> |
| JSTL Functions | Funciones estándar JSTL | <fn:toUpperCase> <fn:toLowerCase> |
| Facelets | Lenguaje de plantillas | <ui:component> <ui:insert> |
| JSF HTML | Componentes estándar JSF basados en HTML | <h:body> <h:inputText> |
| JSF Core | Componentes específicos de JSF | <f:actionListener> <f:attribute> |
| RichFaces | Componentes específicos de RichFaces | <h:dataTable> <rich:panel> <rich:popup |
| Ajax RichFaces | Funciones Ajax de RichFaces | <a4j:commandButton> |

Figura 6. Diagrama de facas de una petición de JSF. [38]

Los Facelets consisten en páginas creadas en XHTML, dentro de las cuales podemos utilizar los siguientes elementos:

- Librerías de tags de Facelets, JSF y JSTL: Se utilizan fundamentalmente para crear los diferentes componentes de la interfaz de la página, además de especificar la forma de validarlos o de presentar los datos.
- Lenguaje de expresiones (EL): Nos permite relacionar los componentes anteriores con datos de nuestra aplicación.
- Plantillas de componentes y páginas: Podemos definir la estructura de contenido de nuestras páginas mediante una plantilla, y aplicar esta plantilla a todas ellas. [39]

3.5 Desarrollo desde cero. Ahora se muestra el proceso de instalación de las herramientas y prueba de dos ejemplos de JSF. Desarrollar una sencilla aplicación JavaServer Faces requiere la realización de estos pasos:

- Desarrollar los objetos del modelo, los que contendrán los datos.
- Añadir las declaraciones del bean controlado al fichero de configuración de la aplicación.
- Crear las páginas utilizando las etiquetas de componentes del Interfaz del Usuario (UI) y las etiquetas "core".
- Definir la navegación entre las páginas. [40]

Componentes necesarios

La creación de aplicaciones basadas en el Modelo-Vista-Controlador se ve facilitada por el uso de frameworks. Un marco de trabajo es el conjunto de interfaces de programación de aplicaciones (APIs) y módulos normalmente acompañados de la documentación y guía de uso que definen la manera de implementar alguna de las capas de la aplicación. La podremos ver también en la estructura o cimiento sobre los que se creará nuestra aplicación.

3.5.1 Librerías de JSF

Tabla 2. Librerías de JavaServer Faces. [41]

Tabla 3. Tabla de prefijos y URL. [41]

| Librería de etiquetas | Prefijo | URI |
|-----------------------|---------|------------------------------------|
| JSTL Core | c: | http://java.sun.com/jsp/jstl/core |
| JSF Faces | f: | http://java.sun.com/jsp/jstl/faces |
| JSF HTML | h: | http://java.sun.com/jsf/htm |
| JSF Core | | |
| RichFaces | | |
| Ajax RichFaces | | |

3.5.2 Instalación de JDK

Descargamos el JDK 8 desde la página oficial <https://www.oracle.com/my/java/technologies/javase/javase8-archive-downloads.html> en este caso se descargó el `jdk-8u202-windows-x64.exe`

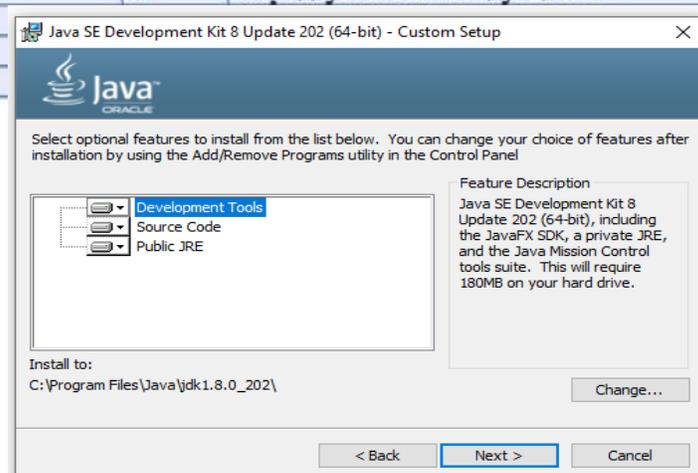


Figura 7. Instalación del JDK con las opciones por default, hasta terminar. Dejamos la configuración por default y pulsamos siguiente, hasta terminar.

3.5.3 Instalación de Eclipse



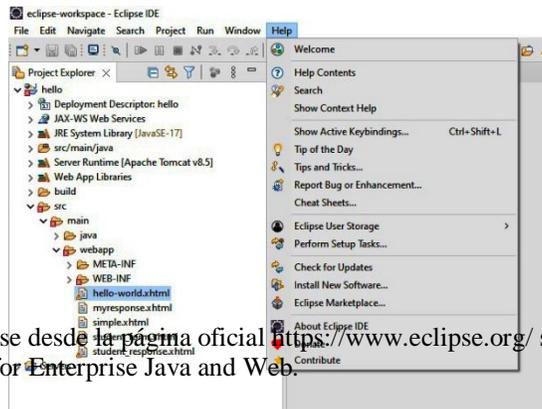


Figura 8. Descarga de eclipse desde la página oficial <https://www.eclipse.org/> se eligió el tipo de Eclipse, el utilizado fue: Eclipse IDE for Enterprise Java and Web.

Se pulsa en aceptar y comienza el proceso de instalación hasta que terminar.

3.4.5 Instalación del servidor de Aplicaciones WEB. Se descarga desde la página oficial <https://tomcat.apache.org/> se descargó la versión 8.5 de Apache Tomcat. Se definen los puertos 8005 y 8080 para apagado remoto y HTTP respectivamente y se define el nombre del servicio: Tomcat8 y las cuentas de administración y contraseñas.

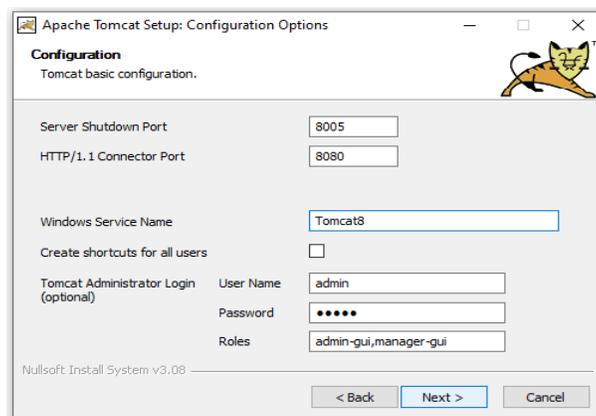


Figura 9. Se definen los puertos 8005 y 8080 para apagado remoto y HTTP.

Comprobamos que la instalación se llevó acabo correctamente ingresando a <http://localhost:8080/>

Conexión de Eclipse y Tomcat. De forma predeterminada, cuando se descarga Eclipse IDE, no viene con Tomcat instalado. Repasemos todos los pasos detallados para configurar Apache Tomcat en un entorno Eclipse.

Ya instalado quedará activo el servidor y aparecerá en el lado izquierdo como servers.

3.6 Desarrollo de ejemplo Hola Mundo

Paso 1. Se crea un proyecto web dinámico

Paso 2. Se nombra el proyecto y se modifica la configuración

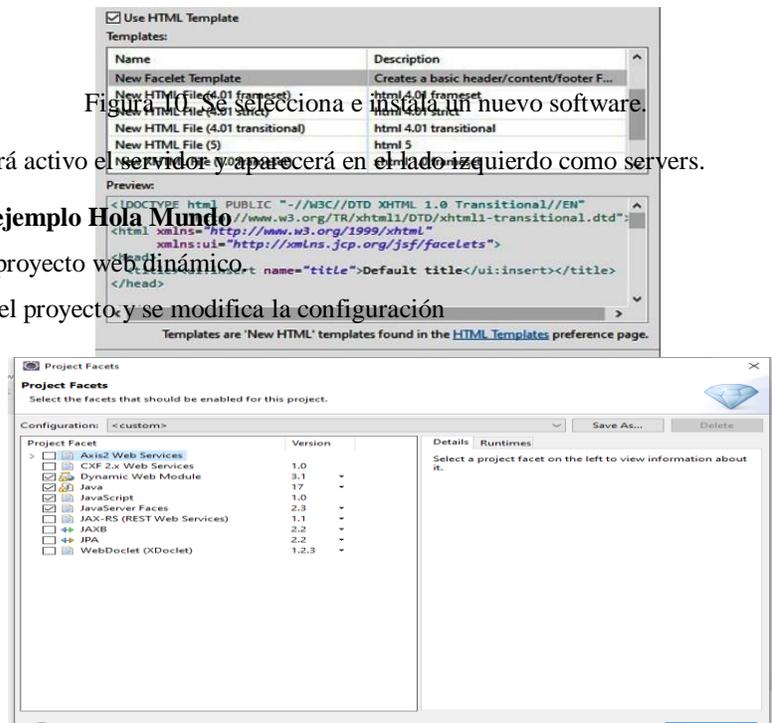


Figura 11. Se marcan los siguientes recuadros: Dynamic Web Module, Java, JavaScript y Java Server Face 2.3 y se pulsa siguiente.

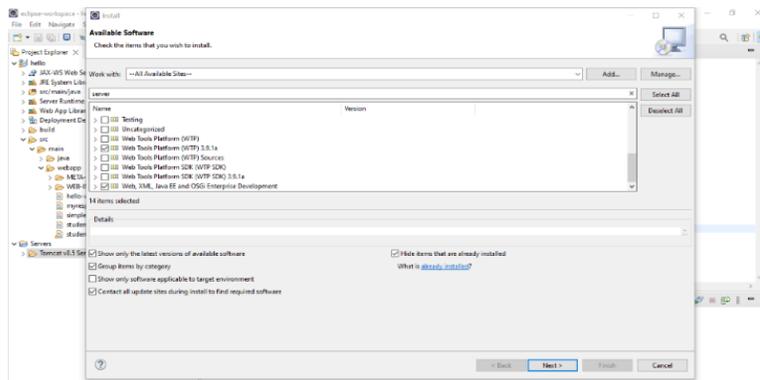


Figura 12. Se marcan los siguientes recuadros del servidor y se continua el proceso.

Paso 3. Se marca el recuadro web.xml del desarrollo del descriptor y pulsamos siguiente.

Paso 4. Se selecciona deshabilitar la configuración de la biblioteca y finalizamos.

Paso 5. Una vez creado el proyecto se crea el archivo HTML.

Paso 6. Se nombra el archivo y se agrega la terminación XHTML.

Figura 13. Se selecciona nueva plantilla de Facelets y se finaliza.

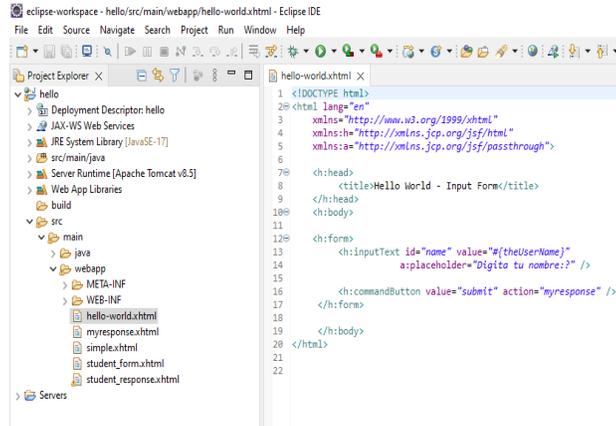


Figura 14. Se sitúa en el archivo XHTML creado y se escribe el código.

Paso 7. Finalmente se ejecuta el programa.

Paso 8. Seleccionamos el servidor, pulsamos siguiente y posteriormente finalizar.

Si el programa es correcto y todas las configuraciones son correctas da como resultado un Hello. Como se muestra en la figura siguiente:

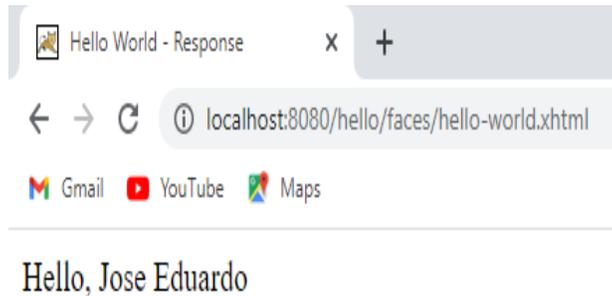


Figura 15. Resultado del proceso.

3.7 Ejemplo con más componentes. Este ejemplo está compuesto de los beans administrados, la página de inicio y la pagina que responde:

Paso 1. Se creó el Beans: Este se crea en la carpeta src/main/java por debajo de JRE System Library.

Paso 2. Con botón derecho del mouse se crea una nueva clase llamada Student, como se observa en la siguiente figura.

El cuadro de la configuración de la clase debe todo por defecto y solo el nombre Student en el campo name.

Paso 3. Agrega la etiqueta **@ManagedBean** y se importa la librería desde: **import**
javax.faces.bean.ManagedBean;

Finalmente se agregó el código siguiente el cual incluye los métodos setter y getters para los atributos y un constructor por default.

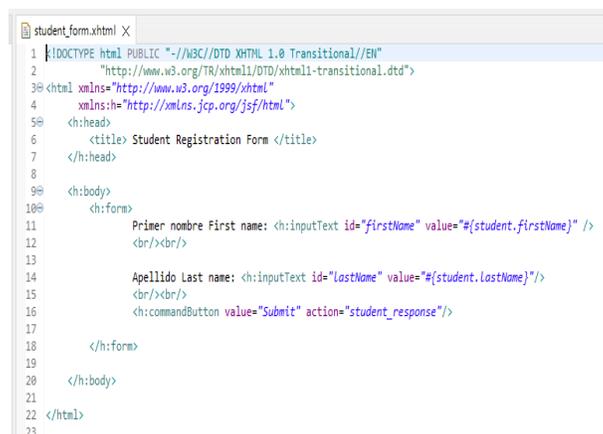
```
package hello2;
import javax.faces.bean.ManagedBean;
@ManagedBean
public class Student {
    private String firstName;
    private String lastName;

    // constructor
    public Student () {

    }
    // se definen los setter y los getters

    public String getFirstName() {
        return firstName;
        //return " ";
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
}
```

Paso 4. Se agrega el archivo XHTML de nombre: student_form.xhtml que contendrá la página de inicio, debe quedar en la ruta src/main/webapp/WEB-INF



```
student_form.xhtml X
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4 xmlns:h="http://xmlns.jcp.org/jsf/html">
5 <h:head>
6 <title> Student Registration Form </title>
7 </h:head>
8
9 <h:body>
10 <h:form>
11 Primer nombre First name: <h:inputText id="firstName" value="#{student.firstName}" />
12 <br/><br/>
13
14 Apellido Last name: <h:inputText id="lastName" value="#{student.lastName}" />
15 <br/><br/>
16 <h:commandButton value="Submit" action="student_response"/>
17
18 </h:form>
19
20 </h:body>
21
22 </html>
23
```

Figura 16. Código creado con métodos setter y getters.

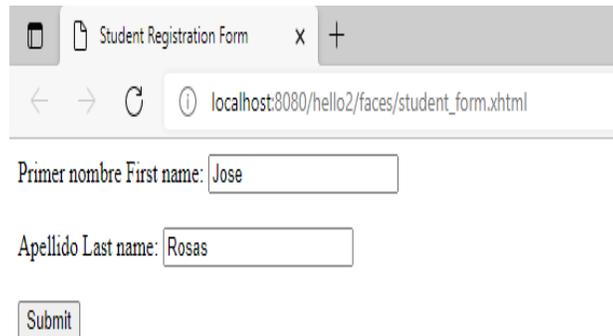
Paso 5. Se agrega el archivo XHTML de nombre: `student_response.xhtml` que contendrá la página que responde a las peticiones recibidas, debe quedar en la ruta `src/main/webapp/WEB-INF`

Los contenidos de ambos archivos XHTML debe ser: **`student_response.xhtml`**

```
student_response.xhtml X
1 <!DOCTYPE html >
2 <html lang="en"
3     xmlns="http://www.w3.org/1999/xhtml"
4     xmlns:h="http://xmlns.jcp.org/jsf/html">
5
6 <head>
7   <title> Student Confirmation </title>
8 </head>
9
10 <h:body>
11   The student is confirmed: #{student.firstName} #{student.lastName}
12 </h:body>
13
14 </html>
15
```

Figura 17. Código para `student_response.xhtml`

Paso 6. Se ejecuta el programa. Y se capturan los datos en la página cuando está en ejecución. Y la página `student_response.xhtml`, responderá como se muestra en la siguiente figura:



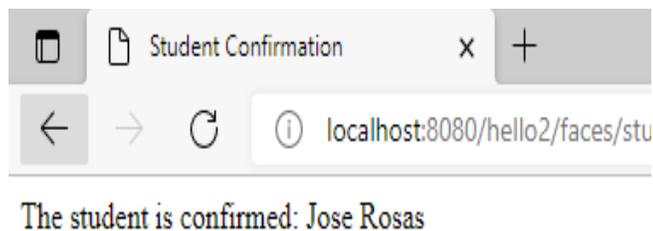
Student Registration Form x +

localhost:8080/hello2/faces/student_form.xhtml

Primer nombre First name:

Apellido Last name:

Figura 18. Se capturan datos en la página resultante



Student Confirmation x +

localhost:8080/hello2/faces/stu

The student is confirmed: Jose Rosas

Figura 19. Resultado del proceso realizado correctamente.

4. CONCLUSIONES.

JavaServer Faces es un framework consolidado para el desarrollo de aplicaciones web dinámicas del lado del servidor orientada a los negocios, que cumple los requerimientos actuales para el desarrollo de tipo enterprise. JSF considera características como seguridad, persistencia de datos, integración con otros sistemas, acceso concurrente y transaccional entre otros. Esta constituido en bibliotecas y librerías de uso libre y de pago. Su programación es de fácil construcción trayendo soluciones a cualquier tipo de problema, ofreciendo una alternativa de bajo costo. JSF está en continua actualización considerando las actuales tecnologías enfocado al mundo de las aplicaciones de negocios

5. REFERENCIAS

- [1]. Julián Pérez Porto y María Merino. (2008). Qué significa WWW. 2021, de Definición.de Sitio web: <https://definicion.de/www/>
- [2]. w3.org. (2014). W3C. 2021, de Ryte Wiki Sitio web: <https://es.ryte.com/wiki/W3C>
- [3]. Julián Pérez Porto y Ana Gardey. (2008). Definición de HTML. 2021, de Definición.de Sitio web: <https://definicion.de/html/>
- [4]. Julián Pérez Porto y Ana Gardey. (2010). Definición de XML. 2013, de Definición.de Sitio web: <https://definicion.de/xml/>
- [5]. Significado.com. (2019). ¿Qué es XHTML? 2022, de Definición.de Sitio web: <https://significa.do/definicion/xhtml/>
- [6]. Rosario Peiró. (2017). Lenguaje CSS. 2022, de Economipedia.com Sitio web: <https://economipedia.com/definiciones/lenguaje-css.html>
- [7]. Redacción. (2021). ¿Qué es JavaScript? 2022, de conceptodefinicion.de Sitio web: <https://conceptodefinicion.de/javascript/>
- [8]. Gonzalo Ramos. (2014). Definición de Java. 2022, de definicion.mx/ Sitio web: <https://definicion.mx/java/>
- [9]. Java jGuru. (2002). Qué es JSP. 2022, de desarrolloweb.com Sitio web: <https://desarrolloweb.com/articulos/831.php>
- [10]. Redacción. (2021). PHP. 2022, de conceptodefinicion.de Sitio web: <https://conceptodefinicion.de/php/>
- [11]. Desarrollo Web. (2001). Qué es ASP. 2022, de desarrolloweb.com Sitio web: <https://desarrolloweb.com/articulos/393.php>
- [12]. Webempresa. (2021). ¿Qué es un servidor Web y para qué sirve? 2022, de webempresa.com Sitio web: <https://acortar.link/s1OYZ9>
- [13]. Webempresa. (2021). ¿Qué es Apache y cómo funciona? 2022, de webempresa.com Sitio web: <https://acortar.link/yBL1oD>
- [14]. Arimetrics. (2021). Qué es Joomla. 2022, de arimetrics.com Sitio web: <https://www.arimetrics.com/glosario-digital/joomla>

- [15]. Guillermo de Jesús Flores Ginés. (2021). Modelo Cliente Servidor. 2022, de weebly.com Sitio web: <https://redespomactividad.weebly.com/modelo-cliente-servidor.html>
- [16]. Guillermo de Jesús Flores Ginés. (2021). Modelo Cliente Servidor. 2022, de weebly.com Sitio web: <https://redespomactividad.weebly.com/modelo-cliente-servidor.html>
- [17]. Andrés Schiaffarino. (2019). Modelo cliente servidor. 2022, de infranetworking.com Sitio web: <https://blog.infranetworking.com/modelo-cliente-servidor/>
- [18]. Wordpress. (2013). Aplicaciones Cliente-Servidor. 2022, de WordPress.com. Sitio web: <https://acortar.link/G64mVx>
- [19]. The Apache Software Foundation. (2013). Apache Tomcat®. 2022, de tomcat.apache.org Sitio web: <https://tomcat.apache.org/>
- [20]. José Huamán. (2018). WildFly – 1. Introducción. 2019, de josehuaman.com Sitio web: <https://josehuaman.com/wildfly-introduccion/>
- [21]. Techtarget. (2021). WebLogic. 2022, de techtarget.com Sitio web: <https://www.techtarget.com/searcharchitecture/definition/WebLogic>
- [22]. Begoña Fontanet. (2016). Java EE. 2022, de fundesem.es Sitio web: <https://www.fundesem.es/bt/publicacion-java-ee-y-el-desarrollo-web-un-enfoque-de-aprendizaje>
- [23]. Luv2code. (2017). JSF Overview. 2022, de luv2code.com Sitio web: https://www.youtube.com/watch?v=KwUAA4L_9AA&t=152s
- [24]. Pavón, Juan. (2009). El Patrón Modelo-Vista-Controlador (MVC). 2016, de Universidad Complutense Madrid Sitio web: <https://espejodeantagno.files.wordpress.com/2016/04/modelo-vista-controlador-mvc.pdf>
- [25] Junto de Andalucía. (2021). Capa de Presentación. 2022, de juntadeandalucia.es Sitio web: <https://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/arquitectura/capa-presentacion>
- [26] Juan Manuel Barrios. (2001). Java Servlets. 2022, de users.dcc.uchile.cl Sitio web: <https://users.dcc.uchile.cl/~jbarrios/servlets/general.html>
- [27] Alejandro Carrillo Torres. (2019). JavaServer Pages. 2022, de <http://dis.um.es/> Sitio web: http://dis.um.es/~lopezquesada/documentos/IES_1415/IAW/curso/UT5/ActividadesAlumnos/jsp4/index.html
- [28]. Depto. Ciencia de la Computación e IA. (2012). Javabeans. 2022, de jtech.ua.es Sitio web: <http://www.jtech.ua.es/j2ee/restringido/cw/sesion07-apuntes.html>
- [29]. Antonio J. Martín. (2018). ¿Qué es JSF? 2022, de docplayer.es Sitio web: <https://docplayer.es/64327309-Java-server-faces-jsf.html>
- [30]. Depto. Ciencia de la Computación e IA. (2013). Introducción a JavaServer Faces. 2022, de jtech.ua.es Sitio web: <http://www.jtech.ua.es/j2ee/publico/jsf-2012-13/sesion01-apuntes.pdf>
- [31]. Wikipedia. (2021). JavaServer Faces. 2022, de wikipedia.org Sitio web: https://es.wikipedia.org/wiki/JavaServer_Faces
- [32]. Dpto. Ciencia de la Computación e IA. (2014). Introducción a JavaServer Faces. 2022, de jtech.ua.es Sitio web: <http://www.jtech.ua.es/j2ee/publico/jsf-2012-13/sesion01-apuntes.html>
- [33]. Cristian Henao. (2017). Componentes básicos Java Server Faces. 2022, de codejavu Sitio web: <https://acortar.link/iFZXqJ>

[34]. Depto. Ciencia de la Computación e IA. (2010). RichFaces: una librería de componentes profesional. 2022, de jtech.ua.es Sitio web: <http://www.jtech.ua.es/j2ee/2008-2009/restringido/jsf/sesion04-apuntes.html>

[35]. ICEsoft Technologies Inc. (2020). ICEfaces. 2022, de icesoft.org Sitio web: <http://www.icesoft.org/java/projects/ICEfaces/overview.jsf>

[36]. Pompa Rodríguez Leandro. (2015). Primefaces. 2022, de ecured.cu Sitio web: <https://www.ecured.cu/Primefaces>

[37]. Depto. Ciencia de la Computación e IA. (2014). El ciclo de vida de JSF. 2022, de jtech.ua.es Sitio web: <http://www.jtech.ua.es/j2ee/publico/jsf-2012-13/sesion03-apuntes.html#:~:text=Lo%20que%20en%20JSF%20se,faces>.

[38]. Marcelo. (2020). JSF Ciclo de Vida. 2022, de jcodepoint.com Sitio web: <https://jcodepoint.com/jsf/ciclo-de-vida/>

[39]. Miguel Ángel Lozano. (2015). Facelets, JSTL y lenguajes de expresiones. 2022, de <http://expertojava.ua.es/> Sitio web: <https://acortar.link/REdTWN>

[40]. Junta de Andalucía. (2021). Manual de JSF. 2022, de [juntadeandalucia.es](http://www.juntadeandalucia.es) Sitio web: <https://acortar.link/YLp3f5>

[41]. Depto. Ciencia de la Computación e IA. (2014). Introducción a JavaServer Faces. 2022, de [jtech.ua.es](http://www.jtech.ua.es) Sitio web: <http://www.jtech.ua.es/j2ee/publico/jsf-2012-13/sesion01-apuntes.html#Implementaciones+y+componentes+JSF>

LISTA DE TABLAS

1. Versiones JavaServer Faces.
2. Librerías de JavaServer Faces.
3. Tabla de prefijos y URL.

LISTA DE FIGURAS

1. Esquema de la plataforma Java EE.
2. Diagrama del Modelo-Vista-Controlador.
3. Funcionamiento de JSF para la generación de una página.
4. Proceso de JSF cuando la página ya está generada.
5. Ejemplo de etiqueta HTML.
6. Diagrama de faces de una petición de JSF.
7. Instalación del JDK con las opciones por default, hasta terminar.
8. Descarga de eclipse desde la página oficial <https://www.eclipse.org/> se eligió el tipo de Eclipse, el utilizado fue: Eclipse IDE for Enterprise Java and Web
9. Se definen los puertos 8005 y 8080 para apagado remoto y HTTP.
10. Se selecciona e instala un nuevo software.
11. Se marcan los siguientes recuadros: Dynamic Web Module, Java, JavaScript y Java Server Face 2.3 y se pulsa siguiente.

12. Se marcan los recuadros del servidor y se continua el proceso.
13. Se selecciona nueva plantilla de Facelets y se finaliza.
14. Se sitúa en el archivo XHTML creado y se procede a escribir el código.
15. Resultado del proceso.
16. Código creado con métodos setter y getters.
17. Código para student_response. Xhtml
18. Se capturan datos en la página resultante.
19. Resultado del proceso realizado correctamente.

