Guía de ejercicios - JEE y Java Server Pages (I)

 ¡Hola! Te damos la bienvenida a esta nueva guía de estudio.

¿En qué consiste esta guía?

La siguiente guía de estudio tiene como objetivo practicar y ejercitar los contenidos que hemos visto en clase.

¡Vamos con todo!

Tabla de contenidos

¡Comencemos!

Actividad guiada: Desarrollo de un sitio Java Web Dinámico

¡Estimado Estudiante, sea muy Bienvenido!, en esta actividad vamos a poner en práctica los aprendizajes previos logrados hasta este momento. El objetivo es desarrollar una actividad donde validaremos nuestros conocimientos en la utilización de **Java Server Pages (JSP) en conjunto con el IDE Eclipse, utilizando para esto el lenguaje de programación Java**.

Para lograrlo, necesitarás aplicar todo lo aprendido en los bloques 1,2 y 3 de esta unidad, (te aconsejamos mantener a mano dicho material en caso de cualquier duda).

En este ejercicio vamos a desarrollar un nuevo proyecto **Java Web Dinámico**, en donde vamos a seleccionar un mes del año por pantalla, para que luego nuestro sistema responda con algunas efemérides asociadas a dicho mes.

Para esto nos apoyaremos tanto de la utilización de JSP como elemento de visualización de información, como también los Servlets de Java para la captura de información y retorno de valores respecto de la selección de usuario.

Aplicando los conceptos y herramientas aprendidas hasta ahora, generamos lo siguiente:

1. Un proyecto Java Web Dinámico llamado **"ActividadJSP".**
2. Un JSP llamado **"index.jsp"**, encargado de la presentación para captura de valor de mes seleccionado.
3. Vinculación con Servlet **"/ObtenerEfemerides"** para toma de request y posterior proceso de evaluación con entrega de información asociada al mes.
4. Creación de paquetes y clases Java necesarias.
5. Generación de respuesta con valores de acuerdo al mes seleccionado.

¡Suena genial!

Creación de Proyecto

Para crear el proyecto desde Eclipse debemos ir al menú **"File"**, luego seleccionar **"New"** y finalmente elegir desde el listado la opción **"Dynamic Web Project".**

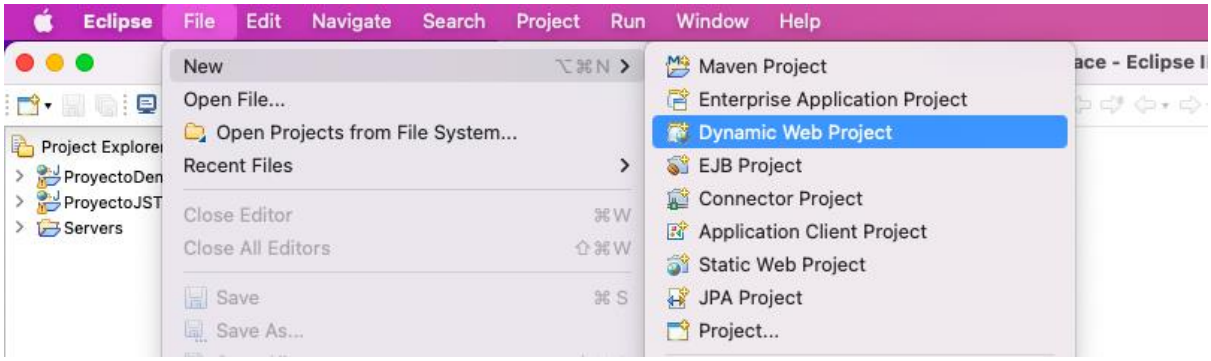

Imagen 1. Creación del proyecto
Fuente: Desafío Latam

Una vez seleccionado, eclipse desplegará el cuadro de creación de un nuevo proyecto. Desde acá debemos indicar el nombre que deseamos usar para el mismo. En nuestro caso le daremos por nombre **"ActividadJSP"**, tal como se muestra en la siguiente imagen:

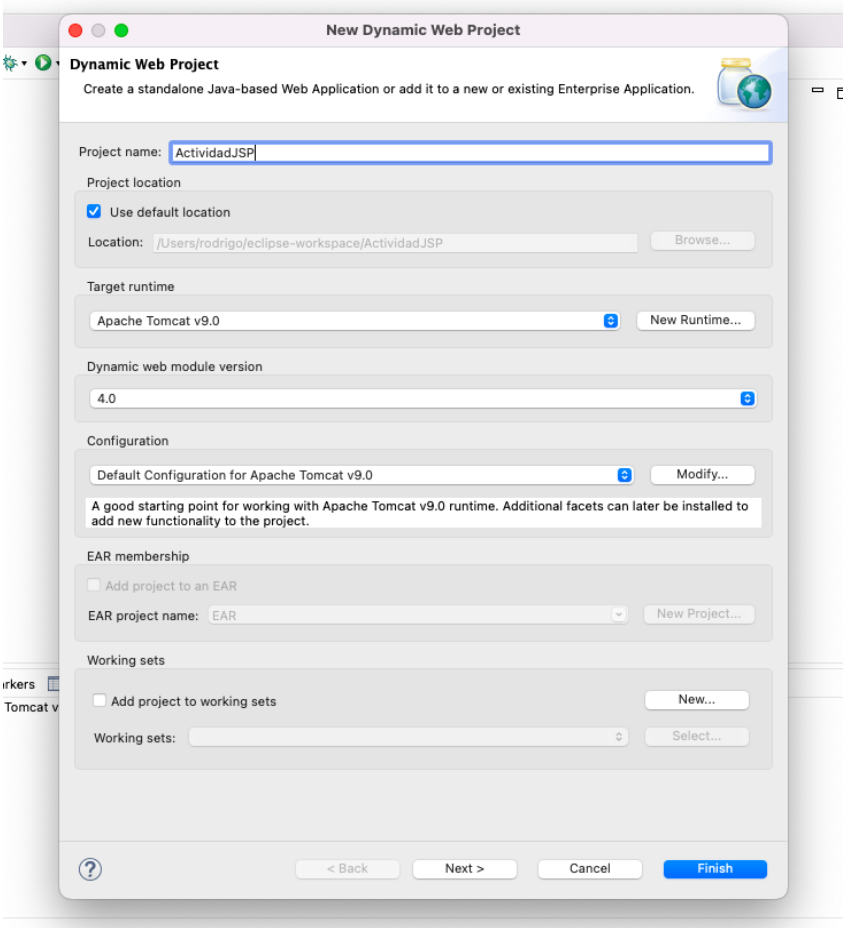

Imagen 2. Creación del proyecto
Fuente: Desafío Latam

Definido el nombre podemos seleccionar otras opciones del proyecto, como lo son la versión de Java y del servidor web.

En nuestro caso, al ya estar todo previamente configurado simplemente dejaremos la opción por defecto que es **Apache Tomcat 9**. (Para volver a revisar el proceso de instalación del servidor web y configuración del IDE recomendamos dar lectura a las presentaciones de la unidad).

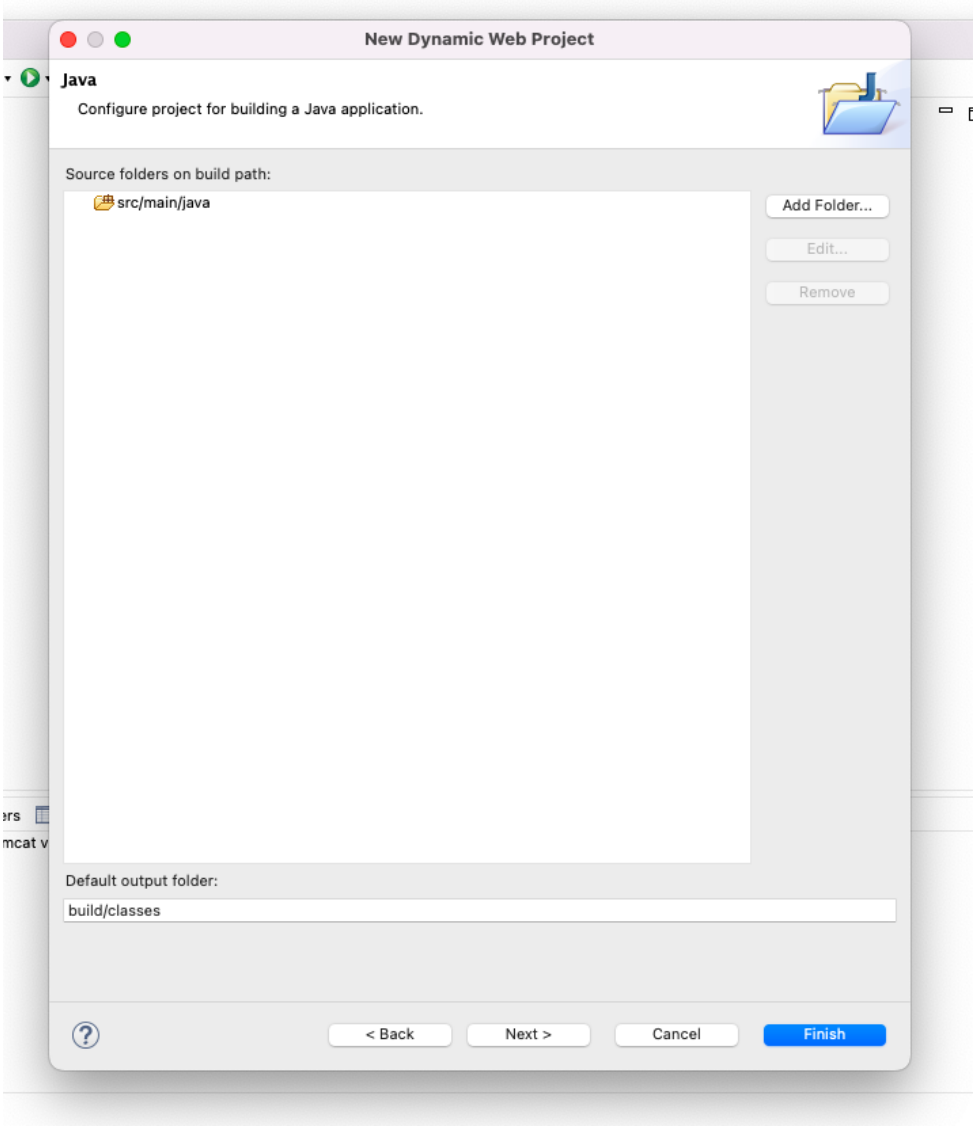

Imagen 3. Creación del proyecto
Fuente: Desafío Latam

Definidas las opciones de creación ya estamos en condiciones de aceptarlas, para tal efecto simplemente damos clic sobre la opción **"Finish"** del asistente.

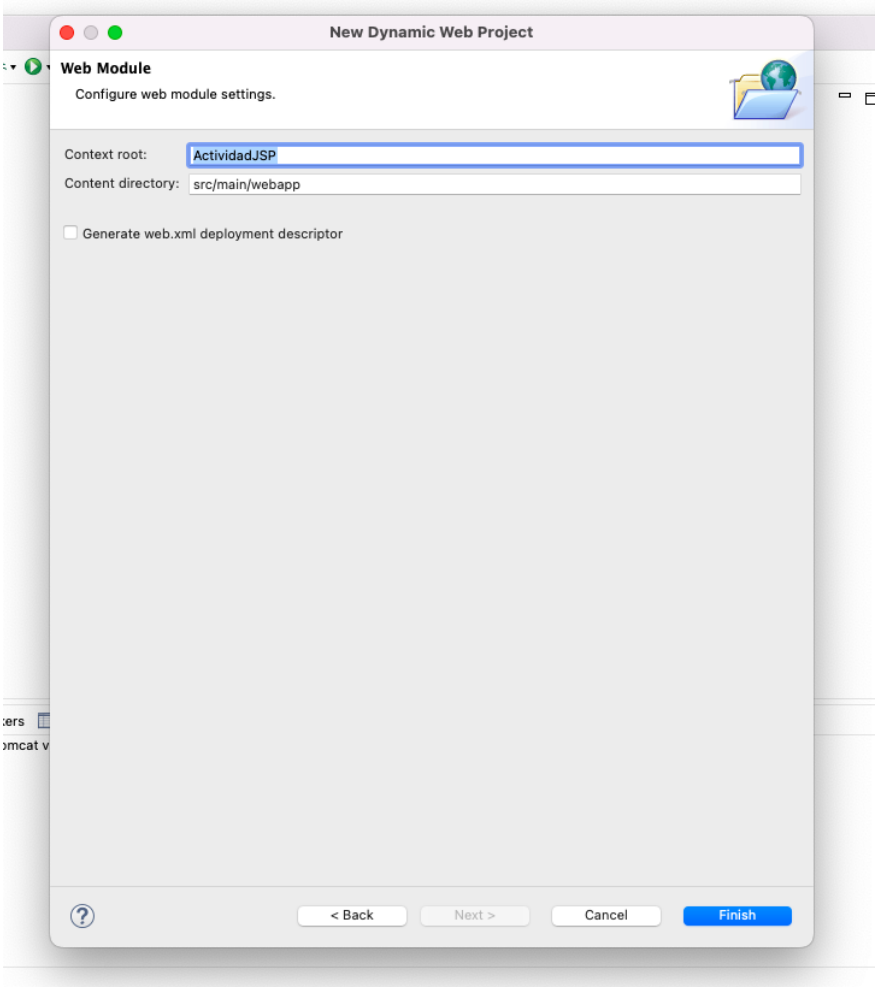

Imagen 4. Creación del proyecto
Fuente: Desafío Latam

Una vez ya creado el proyecto, este aparecerá en la sección de **"Project Explorer" de Eclipse**, lo podemos distinguir fácilmente por su nombre **"ActividadJSP"**. Acá tendremos la estructura completa del proyecto, donde iremos agregando tanto los JSP, Servlets, como clases Java según sea requerido.

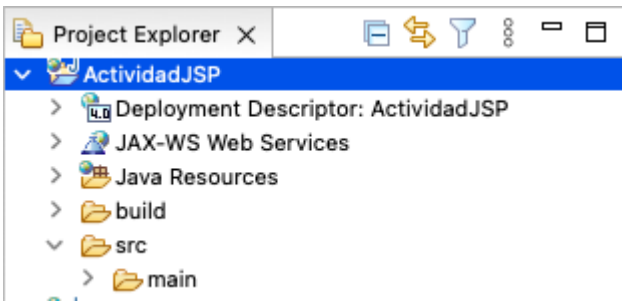

Imagen 4. Creación del proyecto
Fuente: Desafío Latam

Una vez creado nuestro proyecto, el primer elemento que vamos a agregar va a ser nuestro **JSP de entrada al sistema**. Este JSP actuará como elemento index del sitio, y además entregará la información de selección de mes para obtener los resultados deseados.

Para crear nuestro JSP desde el proyecto debemos seleccionar **"File"**, **"New"**, **"JSP File"**.

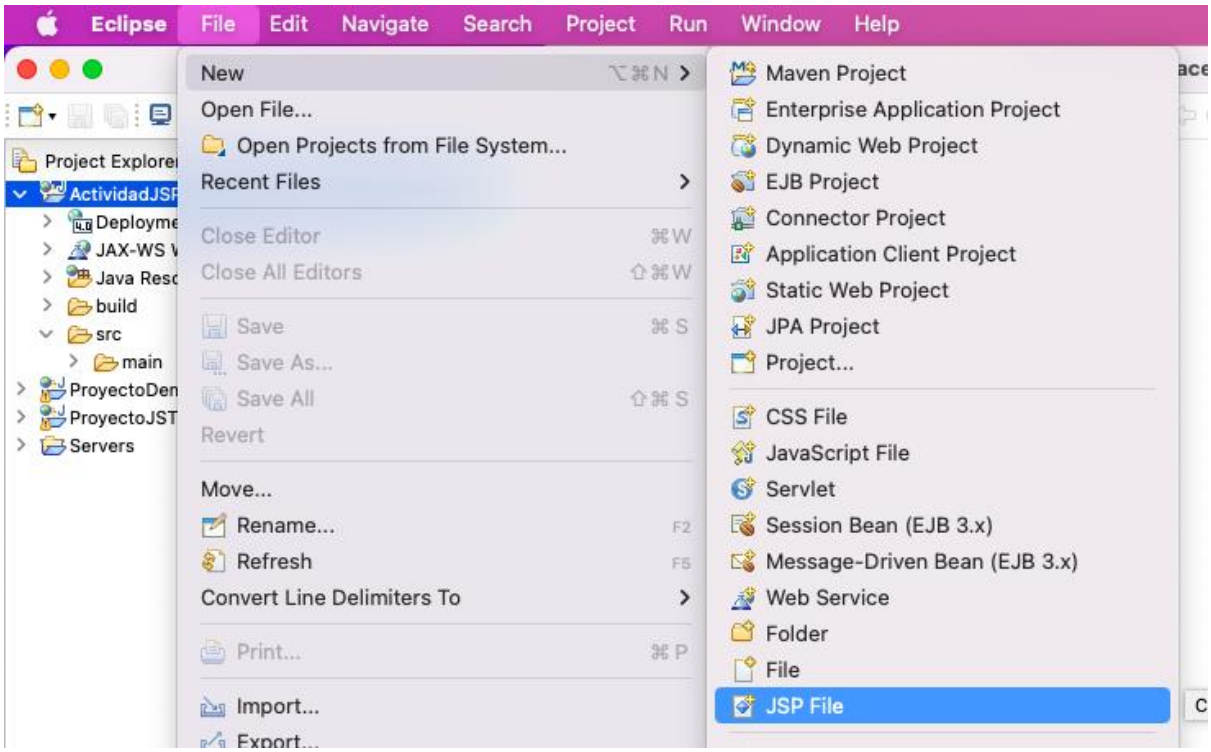

Imagen 5. Creación del JSP
Fuente: Desafío Latam

Una vez seleccionada la opción, Eclipse desplegará el cuadro de creación de nuevo JSP, desde acá debemos indicar el nombre que deseamos entregar al archivo, en nuestro caso el archivo tendrá por nombre **"index.jsp"**.

Importante destacar, que la ruta en la cual estos documentos quedan almacenados en la estructura corresponde a la ruta /webapps

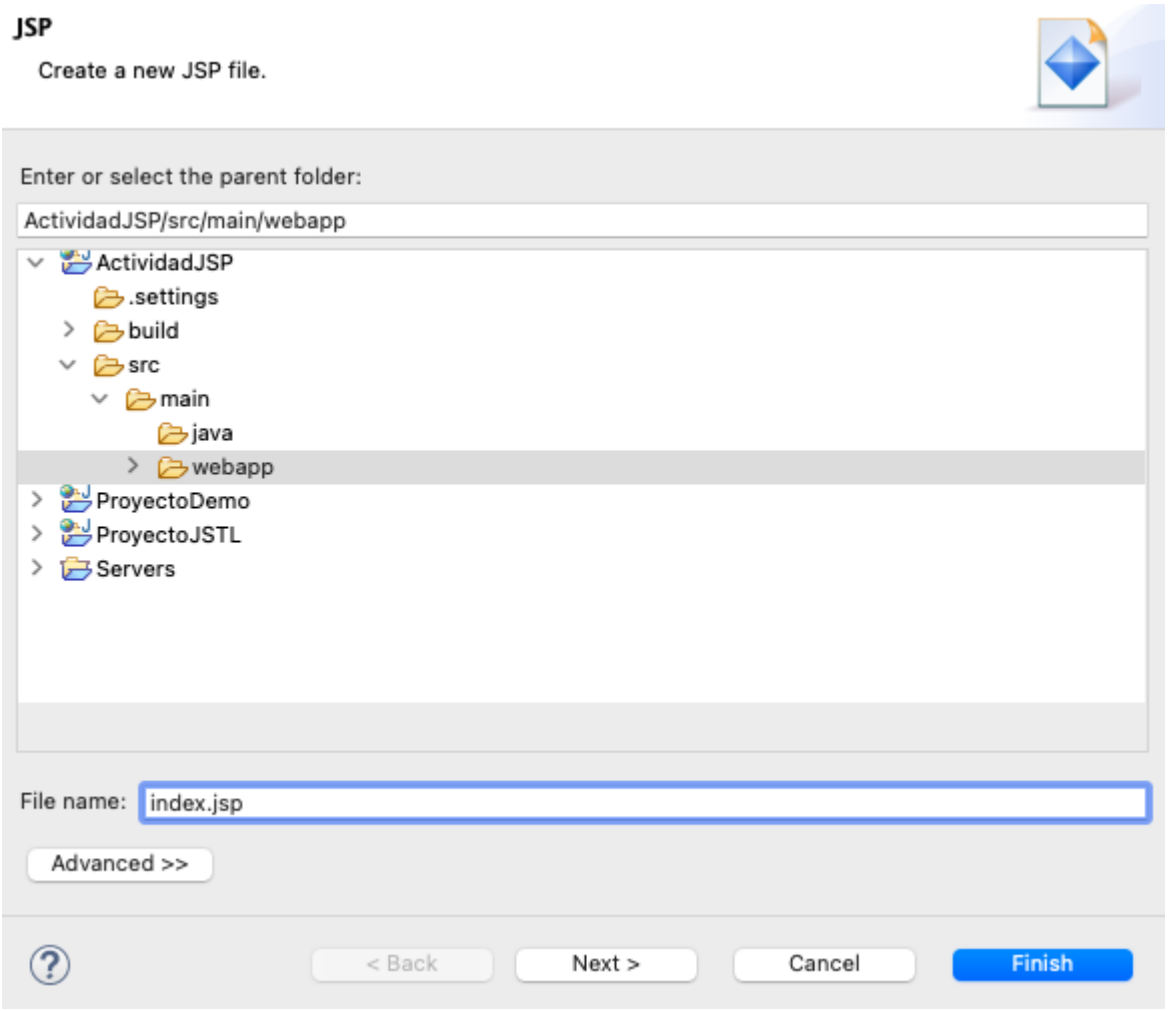

Imagen 6. Creación del JSP
Fuente: Desafío Latam

Una vez ingresado el nombre, simplemente debemos dar clic sobre el botón **"Next" del cuadro de creación de JSP**.

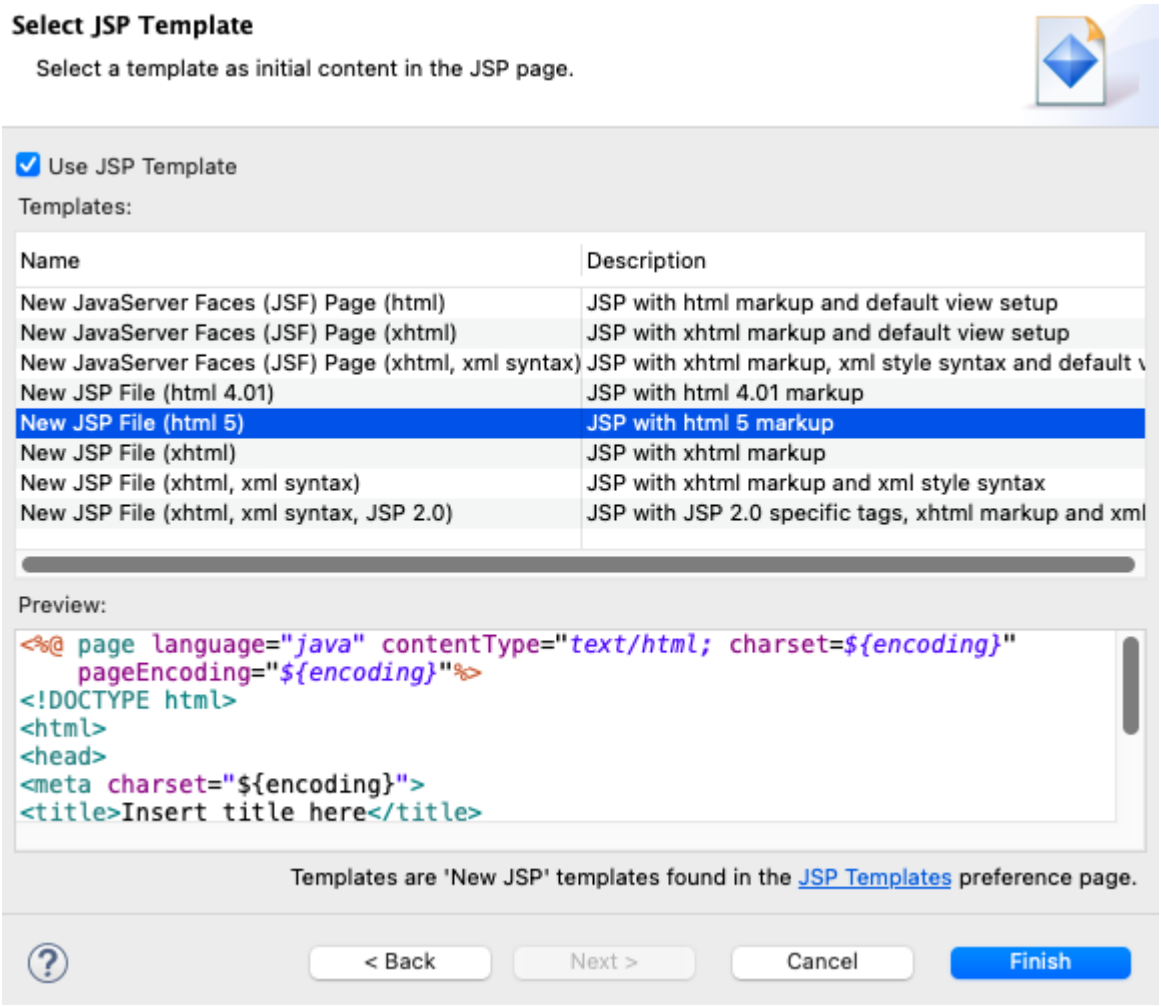

Imagen 7. Creación del JSP
Fuente: Desafío Latam

Luego, podemos definir el **tipo de documento JSP a crear**, en nuestro caso vamos a utilizar el formato de documento por defecto, el cual corresponde a **"New JSP File (HTML 5)".**

Ingresado el nombre ya estamos en condiciones de poder dar clic sobre el botón **"Finish"**, con esto se creará nuestro archivo **"index.jsp".**

IMPORTANTE: Por ahora vamos a dejar nuestro JSP vacío tal y como se entrega por defecto, luego al final del ejercicio volveremos a él para ingresar su contenido.

Creación de Clases

Ya creado nuestro JSP, debemos proceder a definir la estructura de paquetes y clases para el proyecto. En nuestro caso vamos a definir un paquete general llamado **"cl.desafiolatam"**, y sobre este vamos a dejar cada uno de los subpaquetes y/o categorías que necesitemos para cada clase creada.

Para este ejercicio vamos a crear una clase con funciones generales, la cual tendrá métodos generales que usarán tanto los jsp como los servlest, Por tanto, esta quedará dentro del paquete **"cl.desafiolatam.utiles"**.

Desde el proyecto damos clic con el botón derecho y seleccionamos **"New"**, **"Package".**

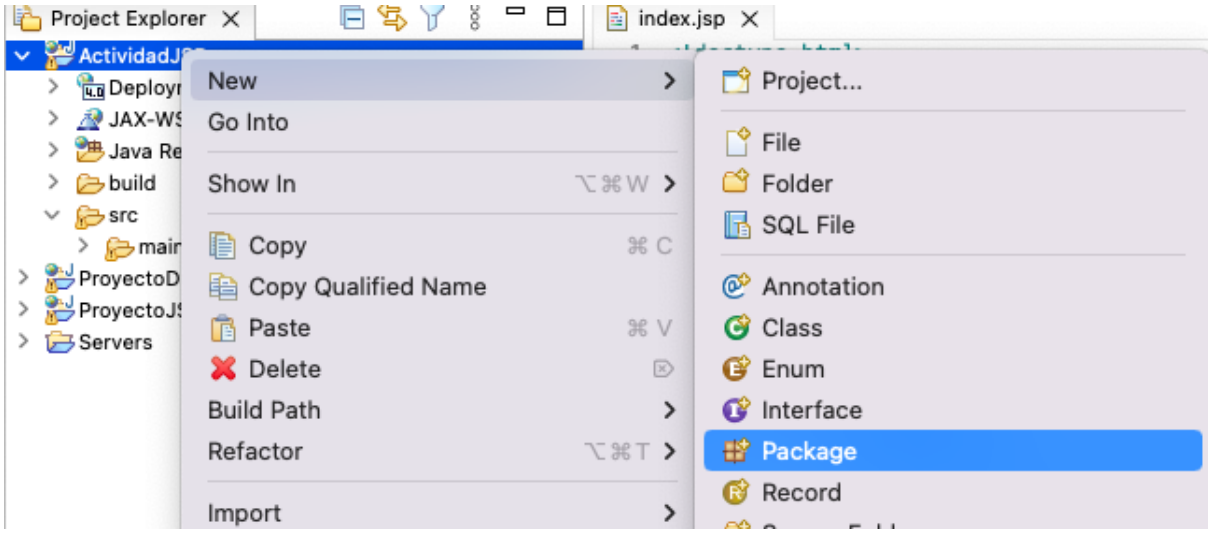

Imagen 8. Creación del Package cl.desafiolatam.utiles
Fuente: Desafío Latam

Luego definimos el nombre del mismo, en nuestro caso el nombre será **"cl.desafiolatam.utiles"**, y finalmente damos sobre el botón **"Finish"**.

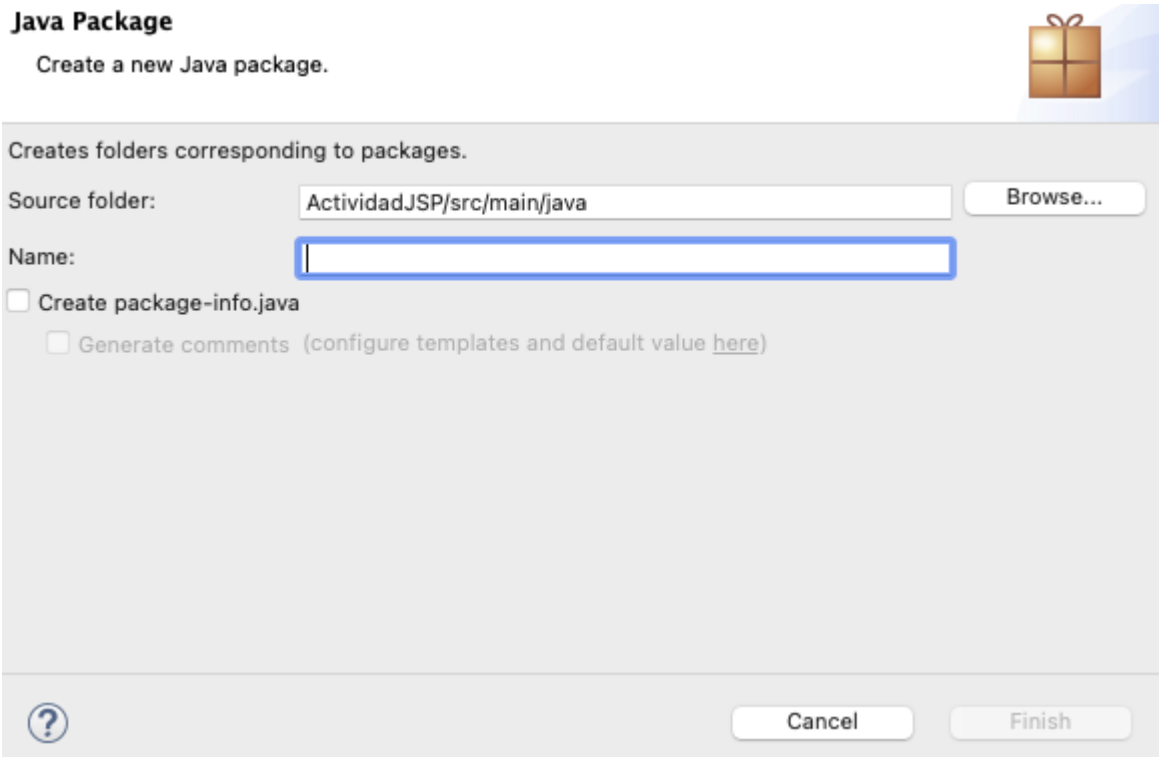

Imagen 9. Creación del Package cl.desafiolatam.utiles
Fuente: Desafío Latam

Una vez ya realizado el proceso anterior, aparecerá en nuestro proyecto el paquete. En este vamos a incluir dentro de él la clase **"UtilesGeneral"**, antes señalada. Para crearla damos clic sobre el nombre del paquete con el botón derecho del mouse y seleccionamos **"New"**, **"Class"**.

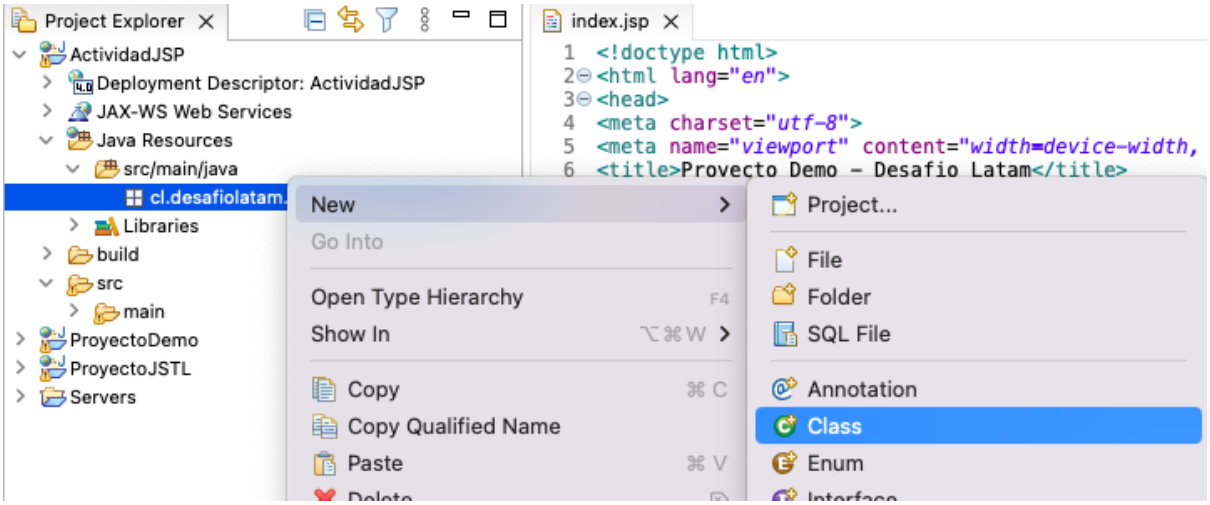

Imagen 10. Creación de la clase UtilesGeneral
Fuente: Desafío Latam

Esto desplegará el cuadro de creación de clases, desde donde podemos definir el nombre de la misma. Por tanto, acá ingresamos en el campo Name el nombre de la clase a crear, la cual es **"UtilesGeneral"**. Importante notar que en el campo Package aparece **"cl.desafiolatam.utiles" (que es el paquete que contendrá esta clase)**, una vez ingresado ya podemos dar clic sobre el botón "Finish".

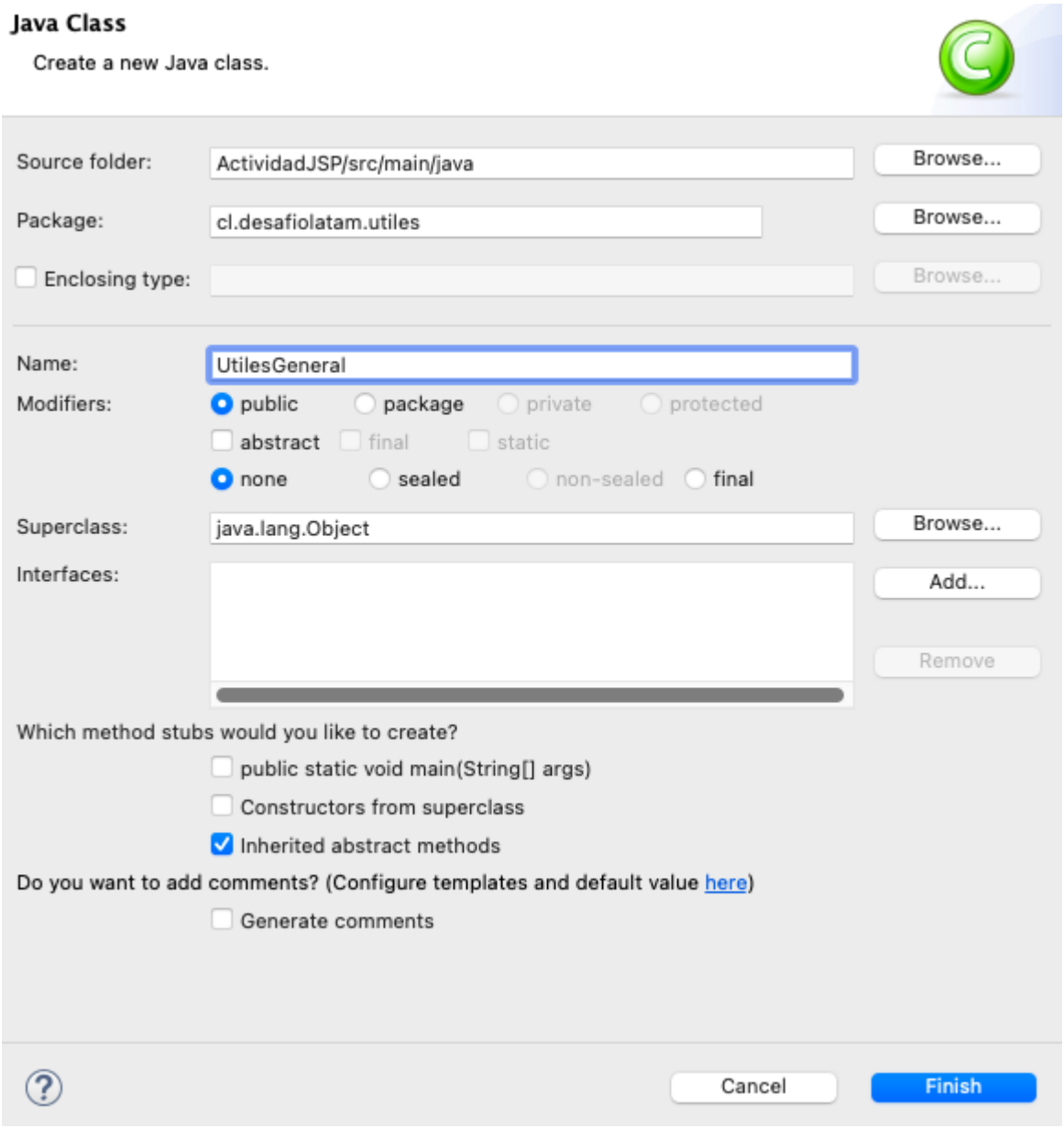

Imagen 11. Creación de la clase UtilesGeneral
Fuente: Desafío Latam

Creada nuestra clase, ya estamos en condiciones de agregar nuestros métodos. El primero será **"getNombreMes"**, utilizado luego desde el jsp para obtener el nombre del mes a partir de un número, y luego el método **"getEfemeridesMes"**, que será utilizado por el servlet para desplegar las efemérides respectivas

Estos métodos han sido simplificados para ser trabajados como arreglos, no obstante, no quita que puedan ser trabajados como clases, listas u otros.

UtilesGeneral.java

```java
package cl.desafiolatam.utiles;

public class UtilesGeneral {


    /**
     * Método que permite obtener el nombre de un mes a partir de su numero
     * @param int numMes, número del mes
     * @return String Nombre Mes
     * @author: Autor <autor@desafiolata.com>
     * @version: 16/08/2022
     */
    public String getNombreMes(int numMes) {

        String[] nombresMeses = { "Enero", "Febero", "Marzo", "Abril",
                "Mayo", "Junio", "Julio", "Agsto", "Septiembre",
                "Octubre", "Noviembre", "Diciembre" };

        // Retorna nombre de Mes
        return nombresMeses[numMes];

    }

}
```

Imagen 12. Creación del Método getNombreMes
Fuente: Desafío Latam

Pegar el contenido del método a insertar en la clase **"UtilesGeneral",** tal como aparece a continuación:

```
package cl.desafiolatam.utiles;

public class UtilesGeneral {

    /**
     * Método que permite obtener el nombre de un mes a partir de su
número
     *
     * @param int numMes, número del mes
     * @return String Nombre Mes
     * @author: Autor <autor@desafiolata.com>
     * @version: 16/08/2022
     */
    public String getNombreMes(int numMes) {

       String[] nombresMeses = { "Enero", "Febrero", "Marzo", "Abril",
"Mayo", "Junio", "Julio", "Agosto", "Septiembre",
                   "Octubre", "Noviembre", "Diciembre" };
```

```
        // Retorna nombre de Mes
        return nombresMeses[numMes];

    }


    /**
     * Método que permite obtener las efemérides de un mes a partir de
su numero
     *
     * @param int numMes, número del mes
     * @return String Efemérides Mes
     * @author: Autor <autor@desafiolata.com>
     * @version: 16/08/2022
     */
    public String getEfemeridesMes(int numMes) {

        String[] efemeridesMes = { "Efeméride 1 Enero, " + "Efeméride 2
Enero, " + "Efemerid 3 Enero",

                "Efeméride 1 Febrero, Efeméride 2 Febrero", "Efeméride
1 Marzo, Efeméride 2 Marzo",
                "Abril - Sin Efemerides", "Efemeride 1 Mayo",
"Efemeride 1 Junio, Efemeride 2 Julio",
                "Efeméride 1 Julio, Efeméride 2 Julio", "Agosto - Sin
Efemerides",
                "Efeméride 1 Septiembre, Efeméride 2, Efeméride 3",
"Octubre - Sin Efemerides",
                "Efemeride 1 Noviembre, Efemeride 2 Noviembre",
"Efemeride 1 Diciembre" };

        // Retorna nombre de Mes
        return efemeridesMes[numMes];

    }

}
```

Ya creado el método, vamos a crear un segundo paquete destinado a almacenar servlets. Para tal efecto, desde el nombre del proyecto damos clic derecho del mouse y seleccionamos **"New"**, **"Package".**

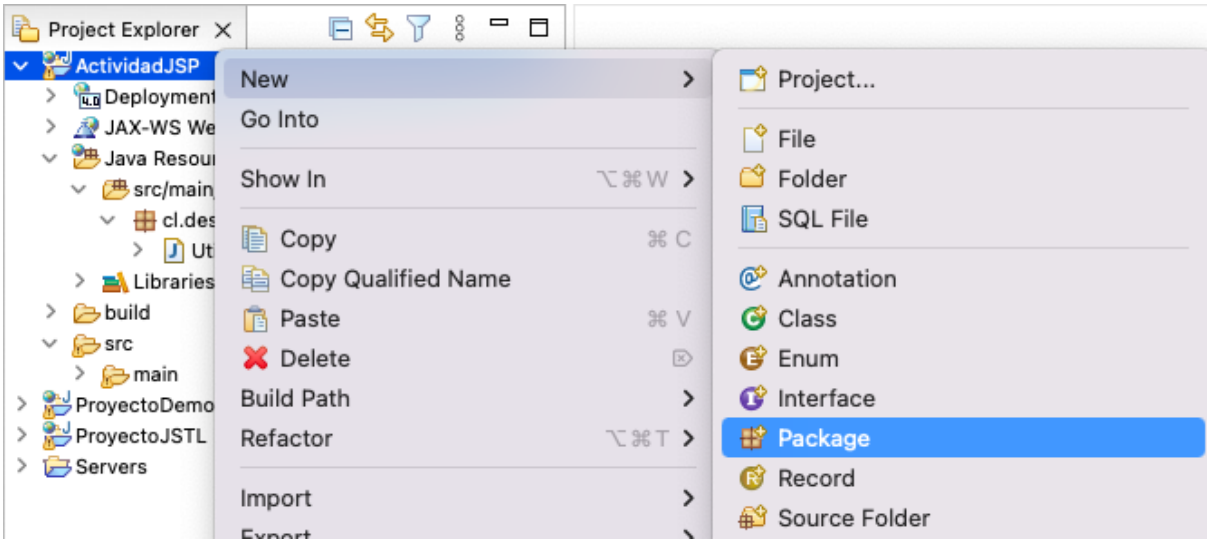

Imagen 13. Creación del package cl.desafiolatam.servlet
Fuente: Desafío Latam

Se desplegará el cuadro de creación de paquetes donde ingresamos el nombre **"cl.desafiolatam.servlet".**

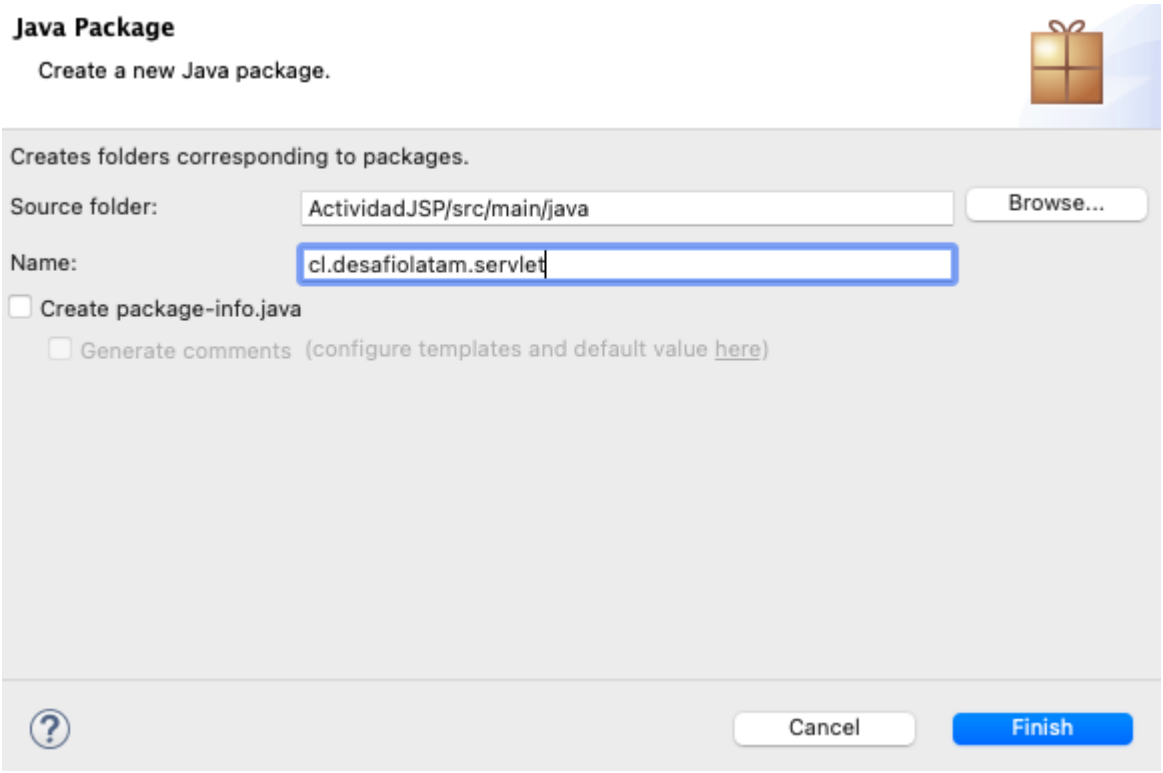

Imagen 14. Creación del package cl.desafiolatam.servlet
Fuente: Desafío Latam

Creado el paquete, este aparecerá en la estructura del proyecto, por tanto, ahora ya estamos en condiciones de agregar un servlet. Para esto sobre el nombre del paquete, presionamos el botón derecho del mouse y luego seleccionamos **"New"**, **"Servlet"**.

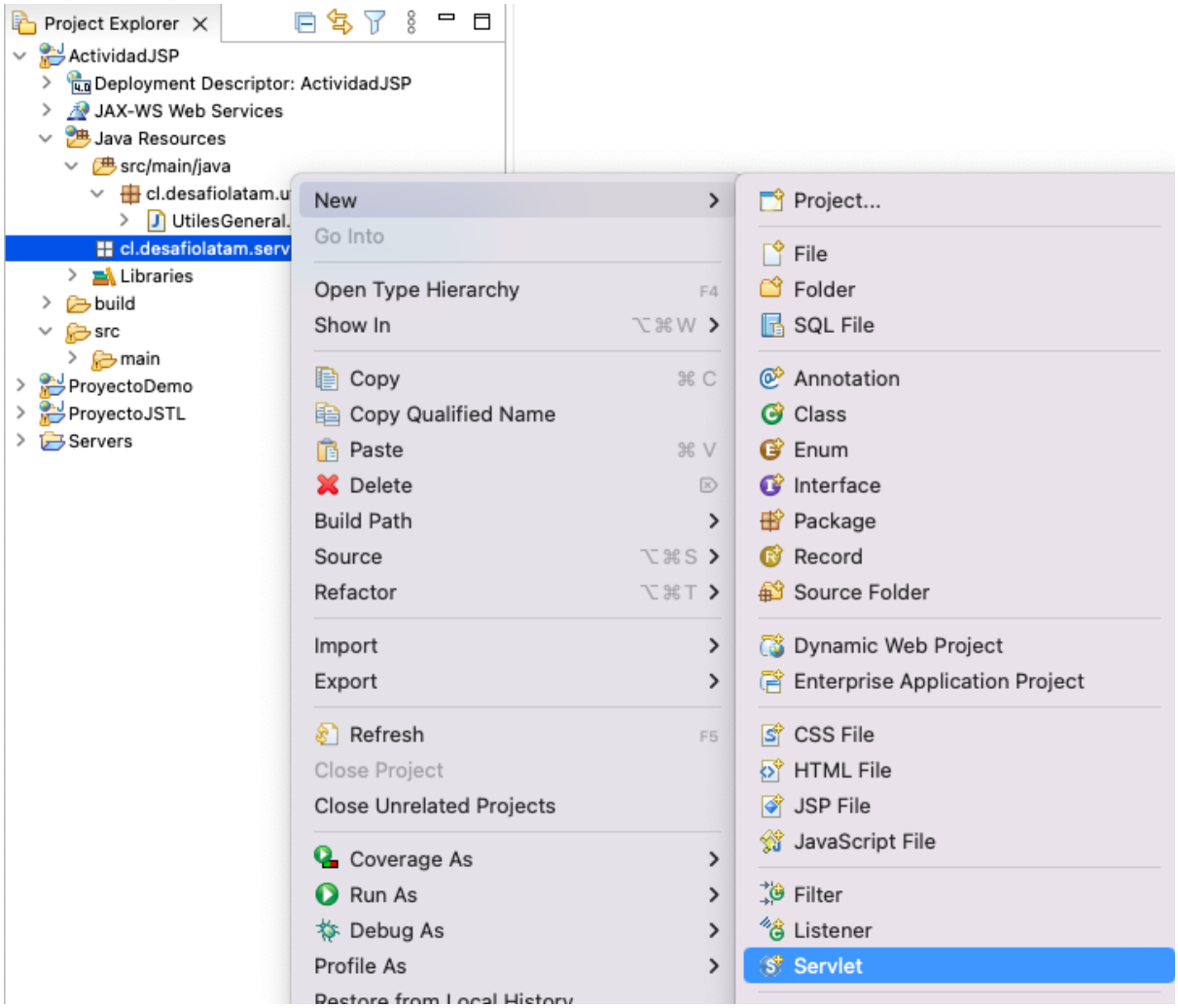

Imagen 14. Creación del servlet ObtenerEfemerides
Fuente: Desafío Latam

Eclipse presentará el cuadro de creación de servlet, donde debemos definir el nombre del mismo, en nuestro caso tendrá por nombre **"ObtenerEfemerides".**

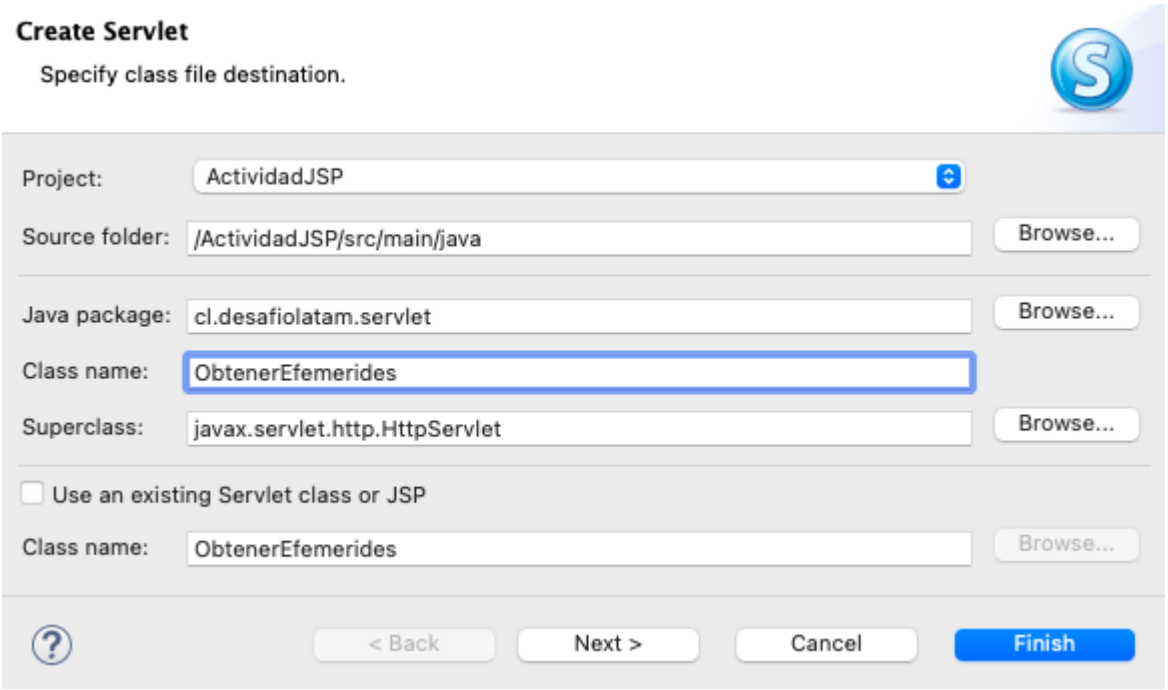

Imagen 15. Creación del servlet ObtenerEfemerides
Fuente: Desafío Latam

Definido el nombre, damos clic sobre el botón **"Next"**, en el cuadro siguiente podemos personalizar el nombre del end-point, pero dado que es un nombre no utilizado lo mantenemos del mismo modo y damos clic sobre **"Next".**

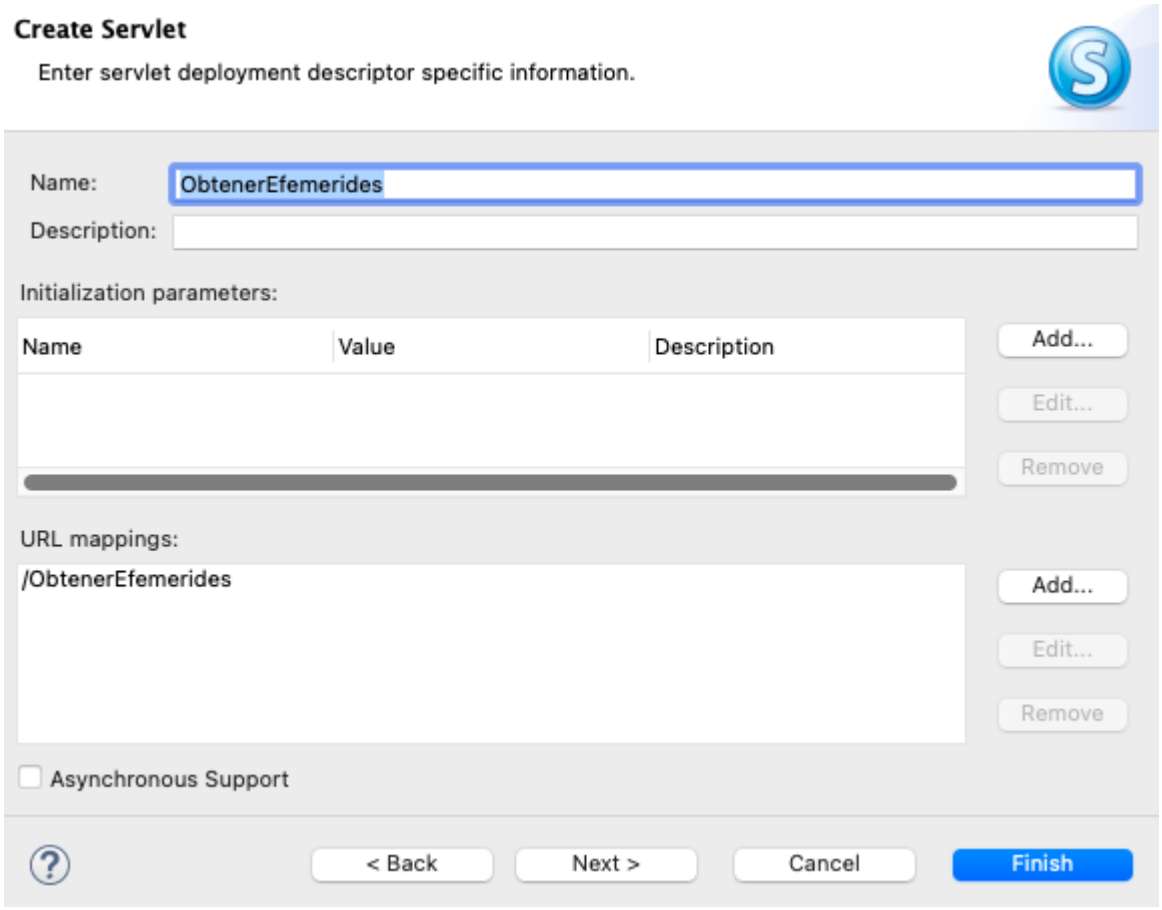

Imagen 16. Creación del servlet ObtenerEfemerides
Fuente: Desafío Latam

Se presentará la info de los métodos por defecto (Post y Get), l**os cuales mantenemos de ese mismo modo,** ya que necesitamos recoger datos desde request y damos finalmente clic sobre el botón **"Finish".**

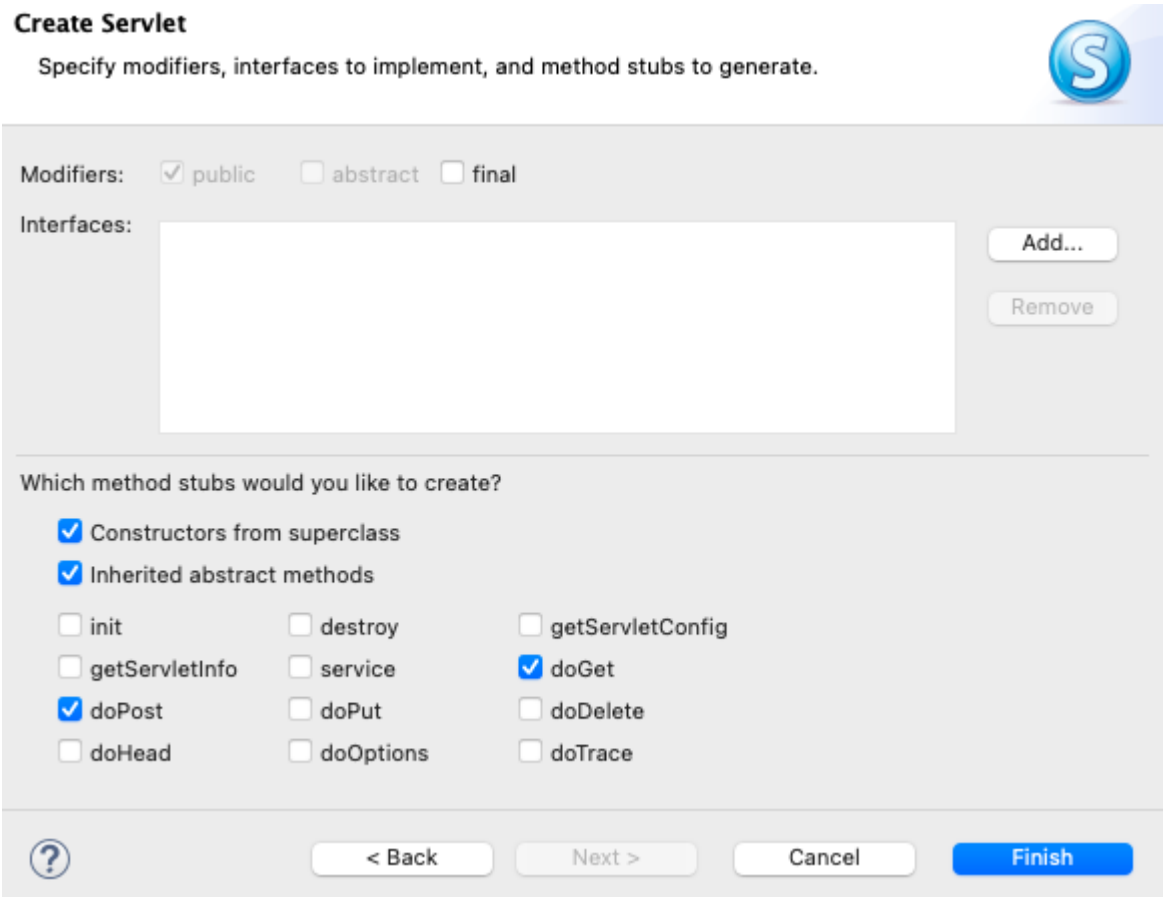

Imagen 17. Creación del servlet ObtenerEfemerides
Fuente: Desafío Latam

Esto procede a crear el servlet respectivo en el paquete **"cl.desafiolatam.servlet"**, por tanto, ahora estamos en condiciones de poder darle lógica de negocio, **! Vamos para allá!**

Contenido Servlet

La lógica de negocio, como se puede ver, está dada por la utilización de la clase **UtilesGeneral** como primer elemento, y luego obtener request desde método doGet, posteriormente de eso se evalúa el mes y se llama al método para rescatar las efemérides correspondientes.

```
package cl.desafiolatam.servlet;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
```

```
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import cl.desafiolatam.utiles.UtilesGeneral;

/**
 * Servlet implementation class ObtenerEfemerides
 */
@WebServlet("/ObtenerEfemerides")
public class ObtenerEfemerides extends HttpServlet {
    private static final long serialVersionUID = 1L;

      /**
      * @see HttpServlet#HttpServlet()
      */
      public ObtenerEfemerides() {
      super();
      // TODO Auto-generated constructor stub
      }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
       // TODO Auto-generated method stub
       //response.getWriter().append("Served at:
").append(request.getContextPath());

       int numMes = Integer.parseInt(request.getParameter("mes"));

       UtilesGeneral utilesgeneral = new UtilesGeneral();
       String efemedires =  utilesgeneral.getEfemeridesMes(numMes);

       response.getWriter().append(efemedires);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request,
```

```
HttpServletResponse response) throws ServletException, IOException {
       // TODO Auto-generated method stub
       doGet(request, response);
    }

}
```

Integración JSP

Finalmente, ya con todo realizado, solo queda integrar el JSP como página index de llamada. Esto quiere decir, que debemos desde acá presentar las opciones de meses al usuario y luego asegurarse que cada una de estas opciones de meses tenga el vínculo de llamada al servlet con paso de parámetro, quedando del siguiente modo:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page import="cl.desafiolatam.utiles.UtilesGeneral"%>
<%
// Incluye clases requeridas
UtilesGeneral utilesgeneral = new UtilesGeneral();
%>
<!doctype html>
<html lang="es">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Proyecto Demo - Desafio Latam</title>
<link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.mi
n.css"
    rel="stylesheet"
    integrity="sha384-
gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx"
    crossorigin="anonymous">
</head>
<body>
    <div class="container">
```

```
	<h1>Actividad JSP - Academia Desafío Latam</h1>
	<div class="row">
		<div class="col-12 col-sm-12">

			<table class="table">
				<thead>
					<tr>
						<th scope="col">#</th>
						<th scope="col">Número Mes</th>
						<th scope="col">Nombre Mes</th>
						<th scope="col">Acciones</th>
					</tr>
				</thead>
				<tbody>
					<%
					for (int x = 0; x < 12; x++) {
					%>
					<tr>
						<th scope="row"><%=x + 1%></th>
						<td>Número Mes <%=x + 1%></td>

<td><%=utilesgeneral.getNombreMes(x)%></td>
						<td><a
href="ObtenerEfemerides?mes=<%=x%>">Ver
Efemérides de
<%=utilesgeneral.getNombreMes(x)%></a></td>
					</tr>
					<%
					}
					%>
				</tbody>
			</table>
		</div>
	</div>

    </div>
    <script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bund
le.min.js"
	integrity="sha384-
A3rJD856KowSb7dwlZdYEkO39Gagi7vIsF0jrRAoQmDKKtQBHUuLZ9AsSv4jD4Xa"
	crossorigin="anonymous"></script>
</body>
```

```
</html>
```

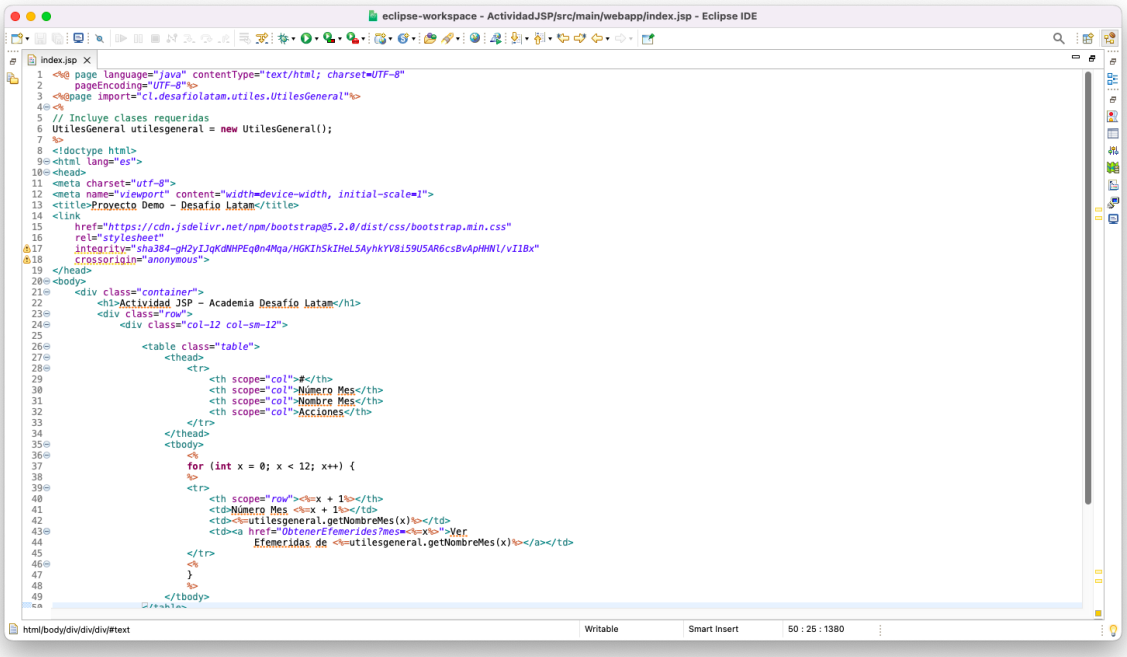

Imagen 18. index.jsp
Fuente: Desafío Latam

Arrancando el Proyecto

Ya realizados todos los pasos, podemos desplegar nuestro proyecto, para esto el proceso es transferir al servidor web Tomcat.

Sobre el proyecto damos clic con el botón derecho del mouse y seleccionamos **“Run As”**, **“Run on Server”.**

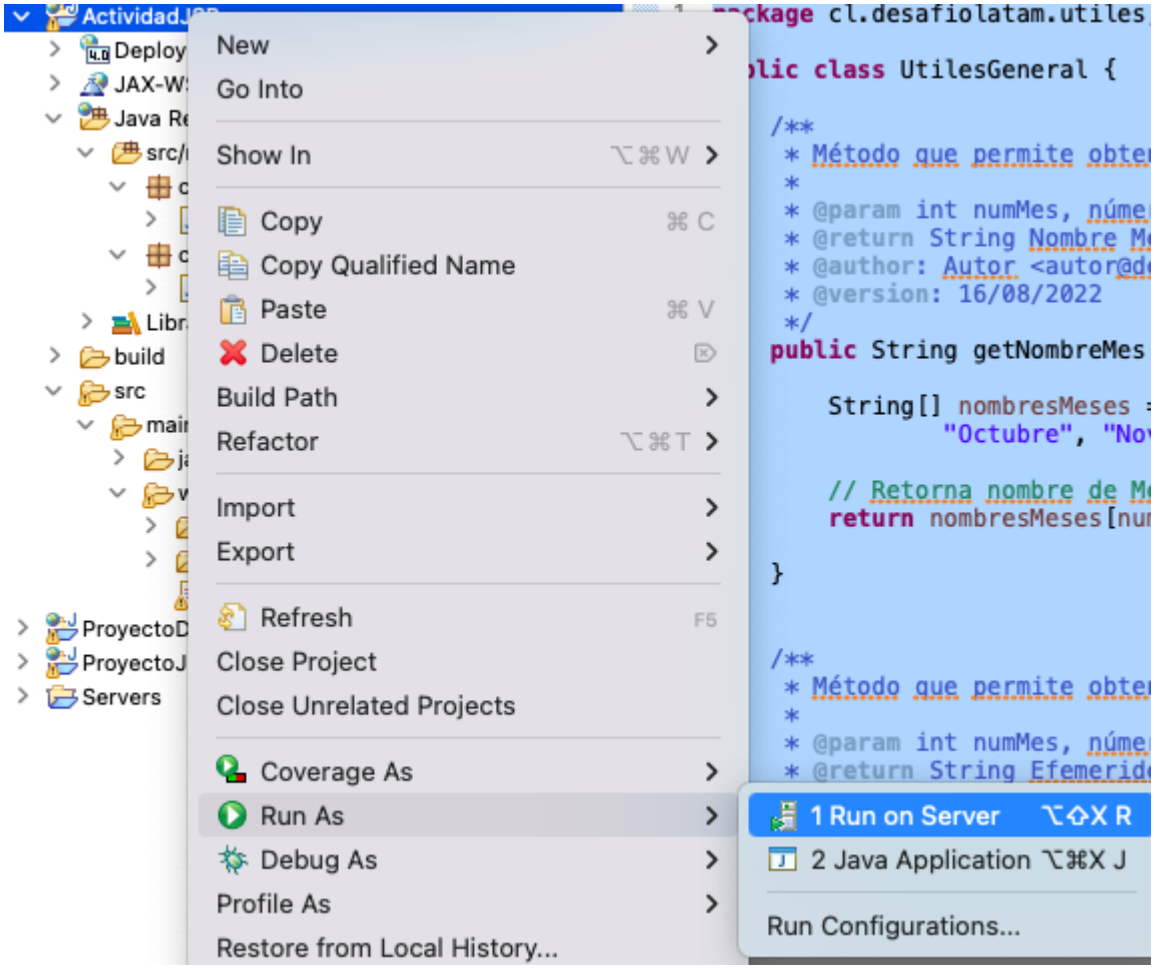

Imagen 19. Arrancando el proyecto
Fuente: Desafío Latam

Se presentará la ventana de ejecución en servidor, desde acá seleccionamos **"Apache Tomcat v.9".**

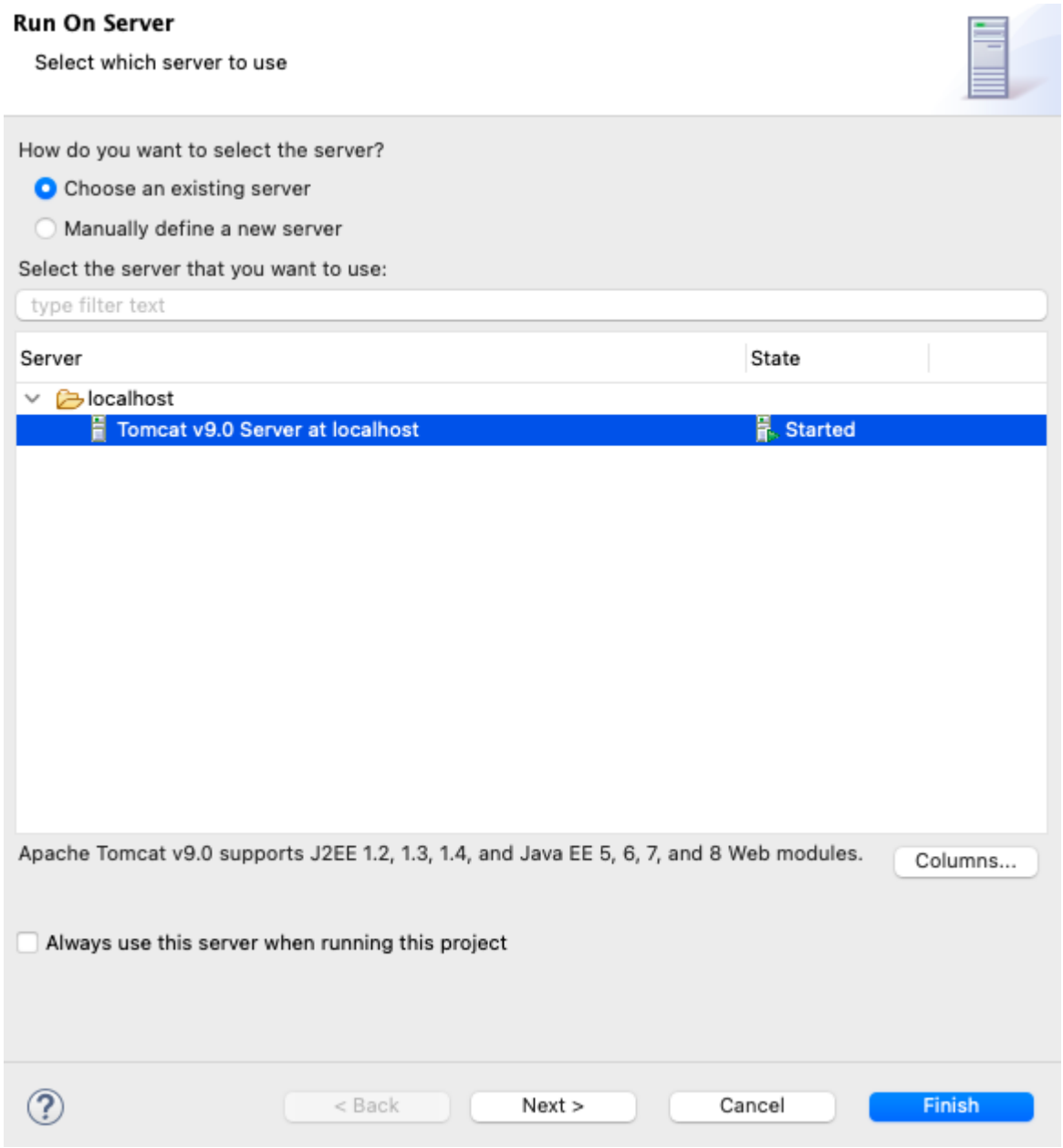

Imagen 20. Arrancando el proyecto
Fuente: Desafío Latam

Debemos desplazar al costado derecho de ejecución el proyecto, para esto seleccionamos el nombre del mismo **"ActividadJSP"** y luego **"Add"**. Finalmente, ya podemos dar clic sobre el botón **"Finish"**.

Esto arrancará el servidor web en la máquina local y desplegará el proyecto en la URL por defecto en nuestro caso **"http://localhost:8080/ActividadJSP".**

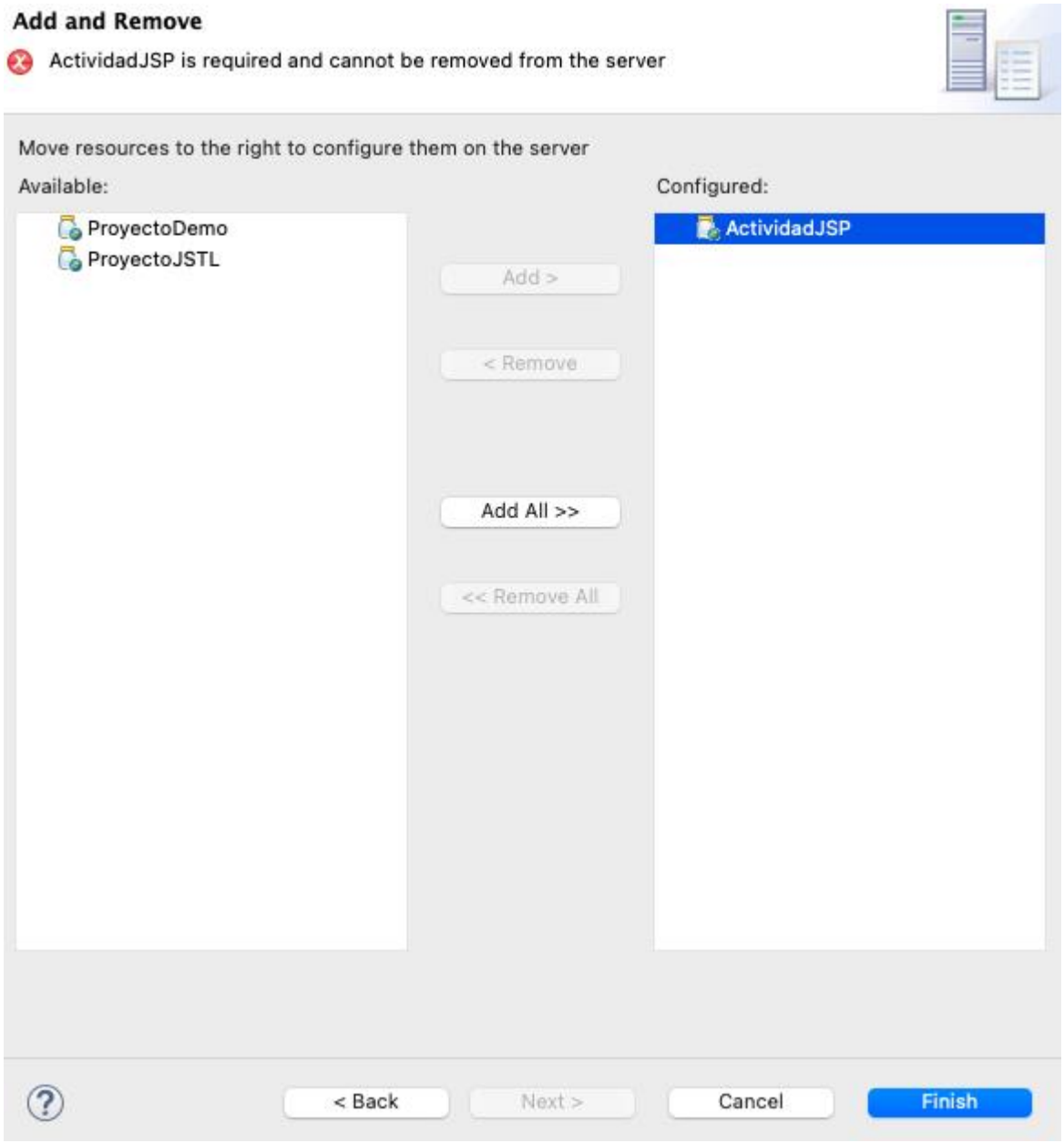

Imagen 21. Arrancando el proyecto
Fuente: Desafío Latam

Proyecto Desplegado

Con lo anterior ya podemos ver nuestro proyecto Java Web Dinámico corriendo y desde el navegador podemos interactuar con sus opciones, tal como se muestra en las siguientes capturas de pantalla.

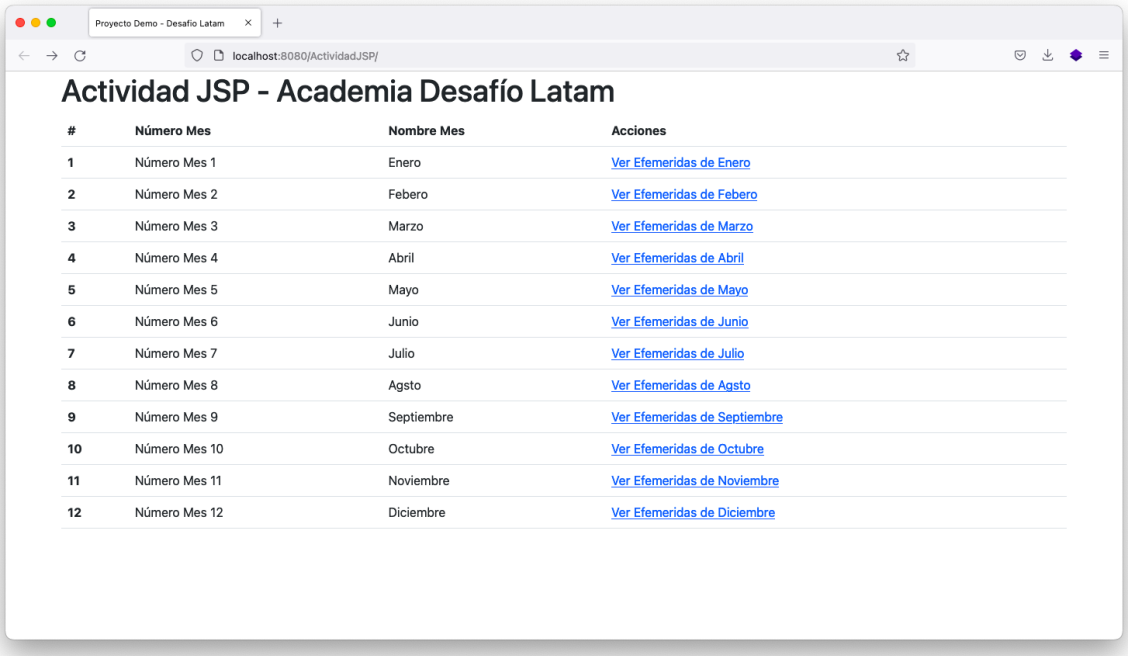

Imagen 22. Pantalla JSP de entrada index.jsp
Fuente: Desafío Latam

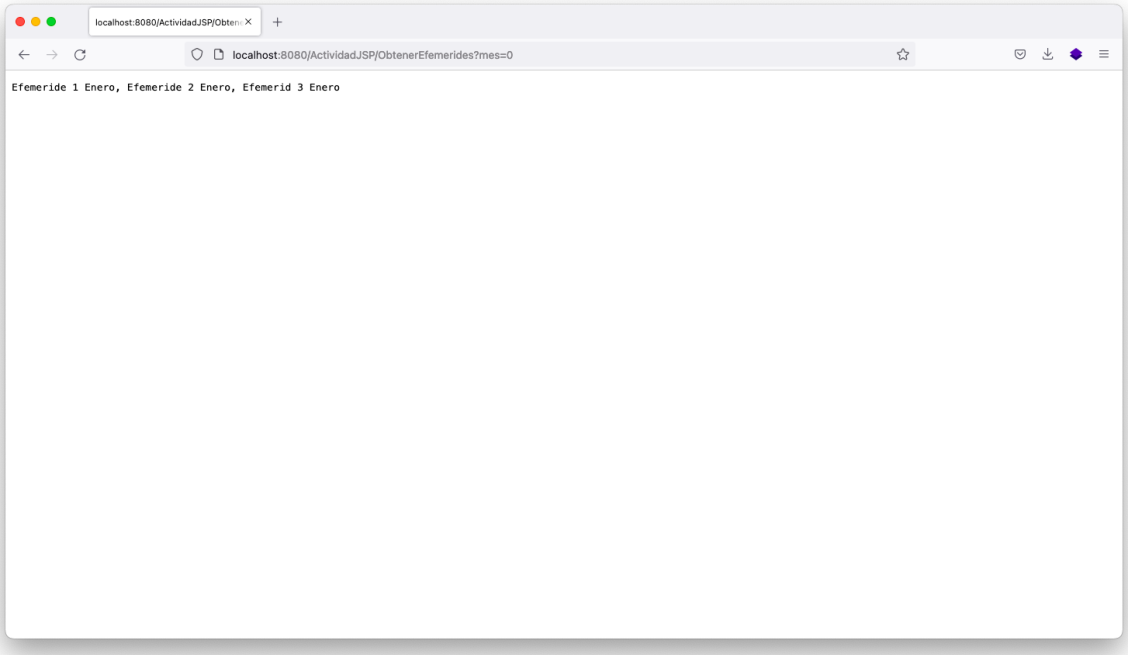

Imagen 23. Pantalla Servlet de Consulta Efemérides
Fuente: Desafío Latam

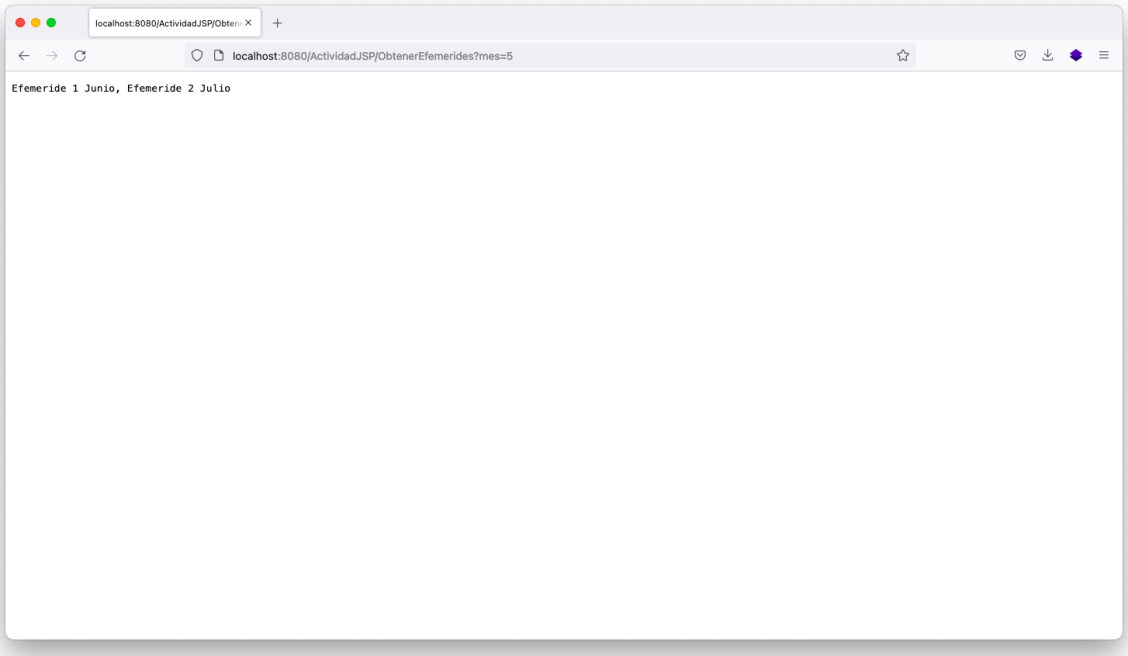

Imagen 24. Pantalla de consulta Efemérides
Fuente: Desafío Latam

¡Manos a la obra!

1.- Obtener cantidad de días del mes

Dado el ejercicio anterior podemos definir una serie de nuevas implementaciones de utilidad, **¡acá la imaginación es nuestro techo!**

Para este caso se propone la creación de una modificación al ejercicio anterior, donde desde el menú principal agregue un segundo llamado a un nuevo servlet llama **"ObtieneDiasMes"**, el cual sea capaz de entregar como respuesta la cantidad de días que dicho mes posee.

Para tal efecto recomendamos agregar en el archivo index.jsp el llamado correspondiente:

```
<a hred="ObtieneMes?mes=1">Mes</a>
```

Donde el servlet posteriormente se encargue de recoger el request y despliegue por medio de una función la cantidad de días para ese mes.

Para esto puedes utilizar arreglos con la cantidad de días del mes, muy similar a como se realizó el ejercicio anterior.

2.- Obtener cantidad de días feriados del mes

Aplicando el mismo concepto anterior, realizar un nuevo ejercicio en donde tengamos un servlet llamado **"ObtieneCantidadFeriadosMes"**, que responda con la cantidad de días feriados que tiene un mes. Recordar para esto usar una función Java, que permita procesar la salida en nuestro servlet.

!Vamos que se puede!

Preguntas de proceso

Reflexiona:

- ¿Qué aspectos nuevos he aprendido del lenguaje de programación Java?
- Respecto del ejercicio desarrollado, ¿hay algo que me haya resultado difícil de desarrollar o entender?
- Sobre servlets y JSP, ¿puedo distinguir la diferencia entre cada uno de ellos?

Preguntas de cierre

- Desde Eclipse, ¿dónde puedo crear un JSP?, ¿dónde puedo crear un servlet?
- ¿Cuál es el proceso para correr una solución web Java desde el IDE Eclipse?
- ¿Cuál es el objetivo de crear paquetes?
- Además de HTML, ¿qué otro tipo de contenido puede tener un JSP?

Referencias bibliográficas

https://www.java.com/es/
https://dev.java/