

Despliegue de aplicaciones web
Practica 2 – Servidor payara
Nombre: Alejandro Freitas Sánchez

Definiendo el contenedor.

1. Mapear los puertos del contenedor con el contenedor.

```
bash CPU: 53% | MEM: 9/17GB 1s 357ms
dockdocker run -d -t -p 2222:22 -p 8080:8080 -p 4848:4848 --name practica2 servidor_payara
9dfb1fffeb4188240b95e83b73a63916191f33ba6859a07415d3cf9a25227882
```

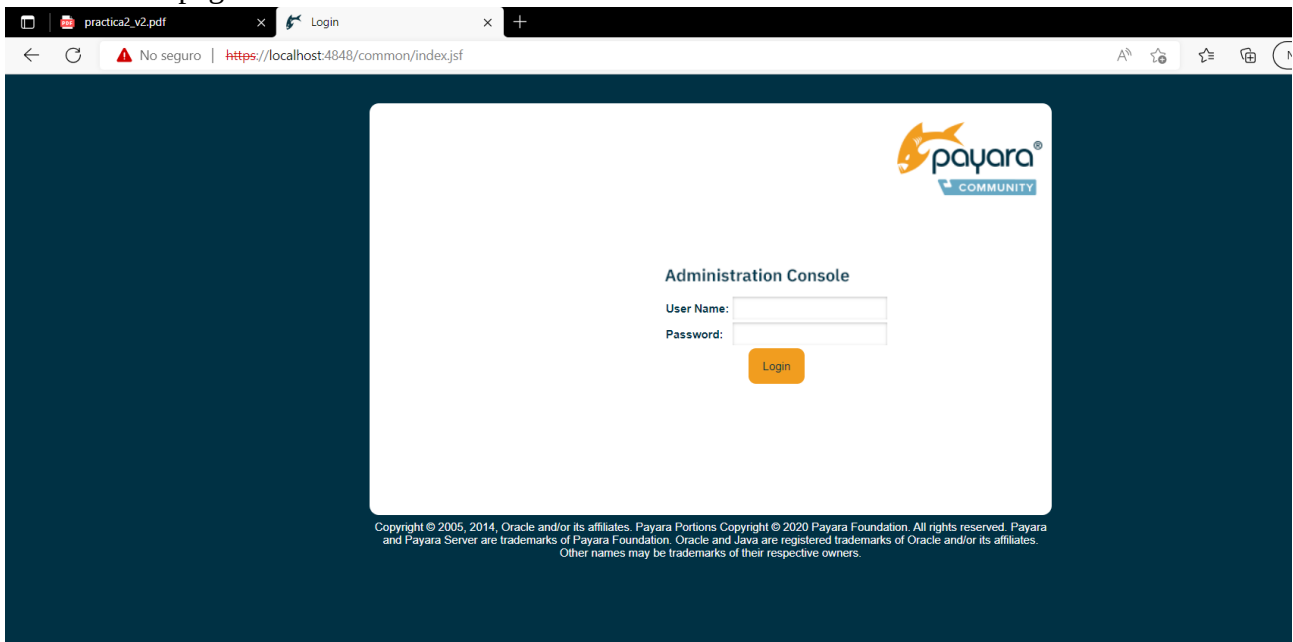
2. Probar en la línea de comandos del contenedor que los path's son correctos.

```
# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
# cd home
# ls
payara5
# cd payara5
# ls
payara5
# cd payara5
# ls
LICENSE.md META-INF NOTICE.md README.txt bin glassfish h2db mq
#
```

3. Probar el ssh.

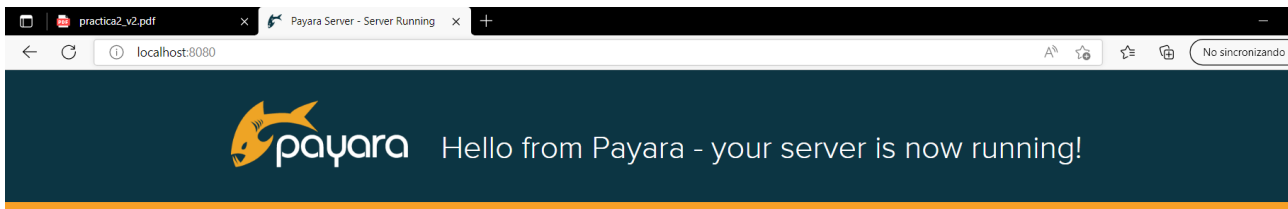
4. Probar el puerto 4848 ¿qué sucede?

Te lleva a la pagina de administración.



5. Probar el puerto 8080.

Lleva a la parte del cliente.



To replace this page, overwrite the file `index.html` in the document root folder of this server. The document root folder for this server is the `docroot` subdirectory of this server's domain directory.

To manage a server on the **local host** with the **default administration port**, [go to the Administration Console](#).

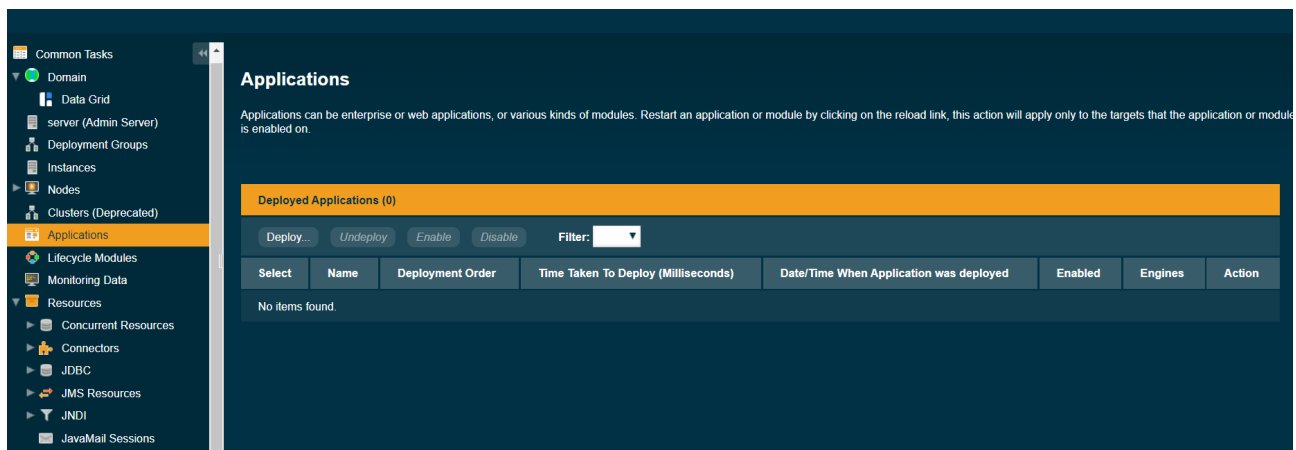
Payara Server Documentation

For more information about Payara Server, documentation and additional resources see the [Payara documentation](#)

Welcome to the Payara Community

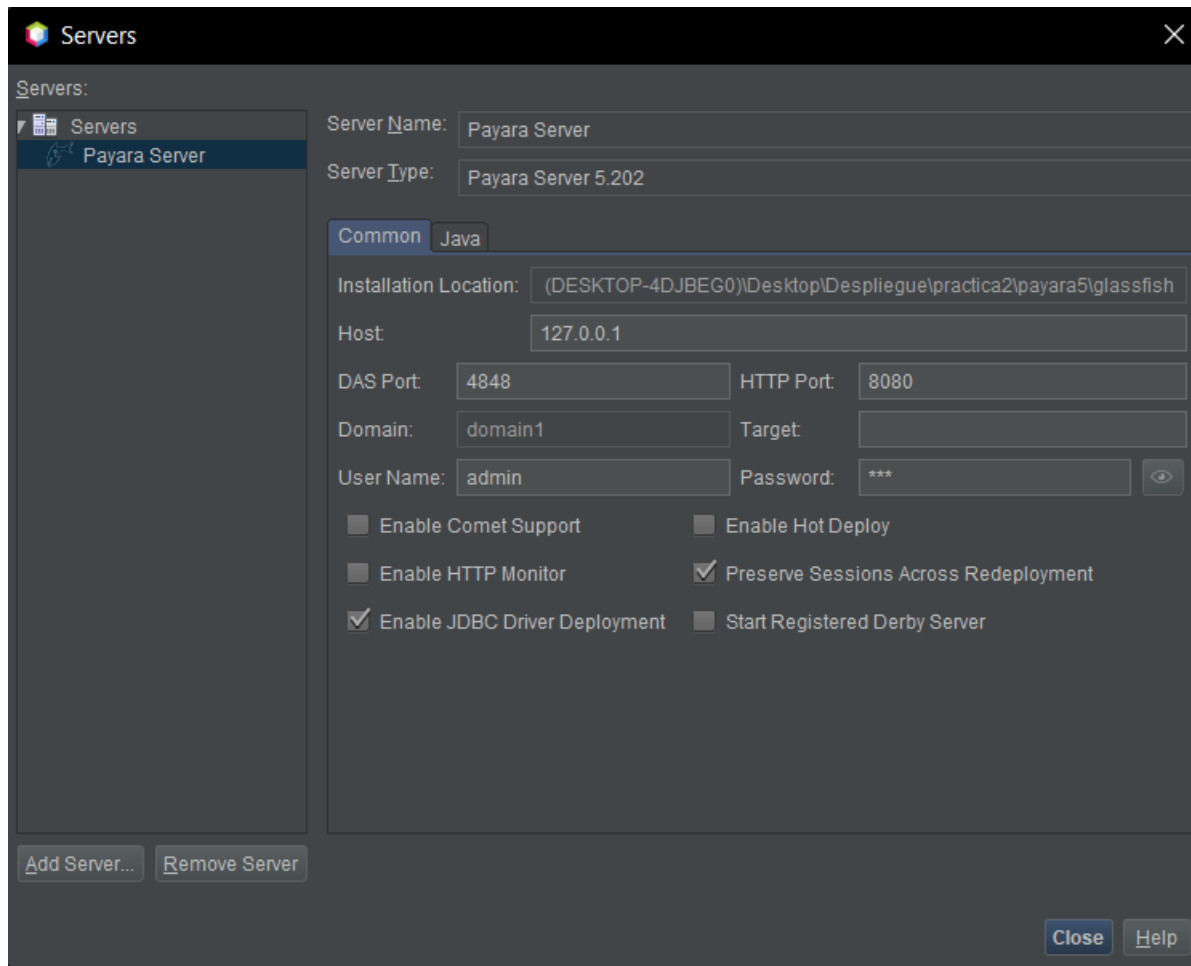
As a member of the continuously growing and evolving open source community, you can benefit from some great support offered by other Payara Server users, and get involved yourself by posting feedback on the latest features and enhancements - or by making suggestions for future releases! And if you want to take it one step further, you can also become an official Payara Contributor. If you'd like to be a part of Payara Server's development, or simply get some help and advice from the community, [check out our Community page](#) for some useful information and links.

6. Ver las aplicaciones instaladas desde el panel de administrador.

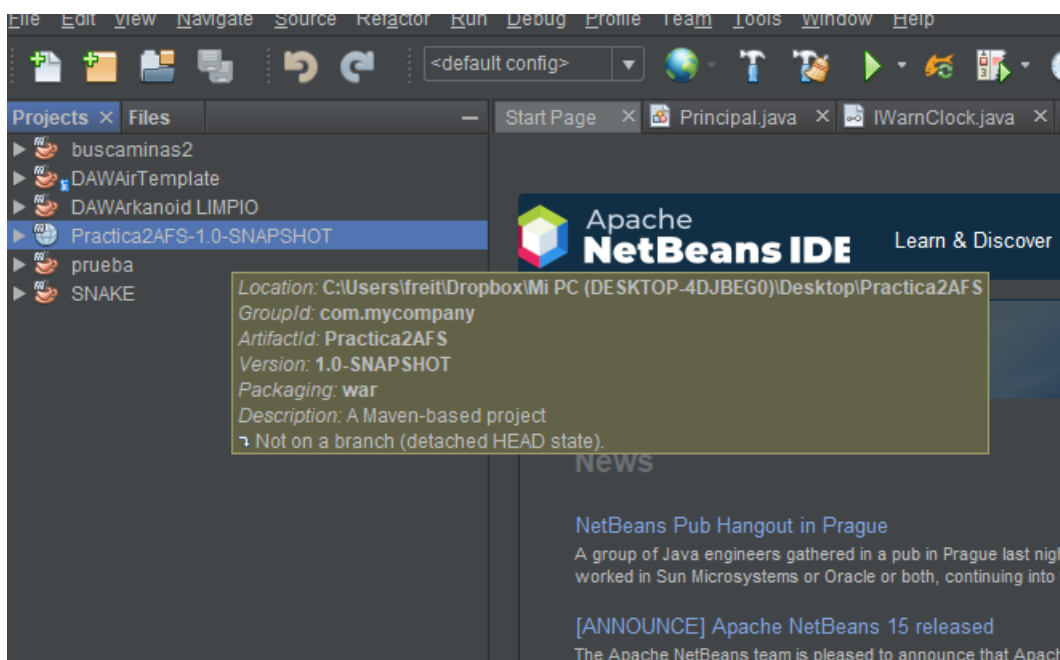


Creando la primera aplicación con Jakarata.

1. En primer lugar registrar el servidor en el IDE, de forma que se pueda realizar el proceso del despliegue de forma rápida. Ir a la pestaña Server y pulsar añadir servidor, seleccionar Payara Server



2. Ahora crear proyecto con Maven de tipo Web Application:



3. Compilar el proyecto. En caso de obtener un error, comprobar la versión del compilador usada (a de ser la 11). ¿Qué tipo de fichero ha generado?

Un archivo .war.

4. Ir a la página web de administración del servidor a la opción Applications, pulsar deploy y subir el fichero war generado. ¿Qué sucede?. Investigar como entrar en la aplicación web.

Con la aplicacion ya añadida, podemos entrar en ella desde launch.

Applications can be enterprise or web applications, or various kinds of modules. Restart an application or module by clicking on the reload link, this action will apply only to the targets that the application or module is enabled on.

Deployed Applications (1)

Deploy... Undeploy Enable Disable Filter:

Select	Name	Deployment Order	Time Taken To Deploy (Milliseconds)	Date/Time When Application was deployed	Enabled	Engines	Action
<input type="checkbox"/>	Practica2AFS-1.0-SNAPSHOT	100	2192	Oct 26, 2022, 12:05:56 PM	✓	health, metrics, openapi, web	Launch Redeploy Reload

5. Eliminar del servidor la aplicación desplegada.

Common Tasks

- Domain
 - Data Grid
 - server (Admin Server)
 - Deployment Groups
 - Instances
 - Nodes
 - Clusters (Deprecated)
 - Applications**
 - Lifecycle Modules
 - Monitoring Data
 - Resources
 - Concurrent Resources
 - Connectors
 - JDBC
 - JMS Resources
 - JNDI
 - JavaMail Sessions

Applications

Applications can be enterprise or web applications, or various kinds of modules. Restart an application or module by clicking on the reload link, this action will apply only to the targets that the application or module is enabled on.

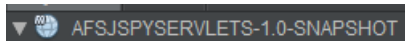
Deployed Applications (0)

Deploy... Undeploy Enable Disable Filter:

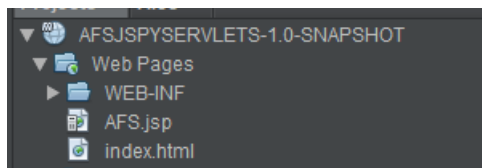
Select	Name	Deployment Order	Time Taken To Deploy (Milliseconds)	Date/Time When Application was deployed	Enabled	Engines	Action
No items found.							

Tecnologías básicas JSP.

1. Crear otra aplicación del mismo tipo llamada TUSINICIALESJSPYSERVLETS.



2. Añade una página JSP de nombre tus iniciales.



3. En el jsp anterior, crear un h1 con tu nombre y hacer que se genere la tabla de multiplica de 10 en el servidor y se muestre en el cliente (investigar cómo hacerlo).

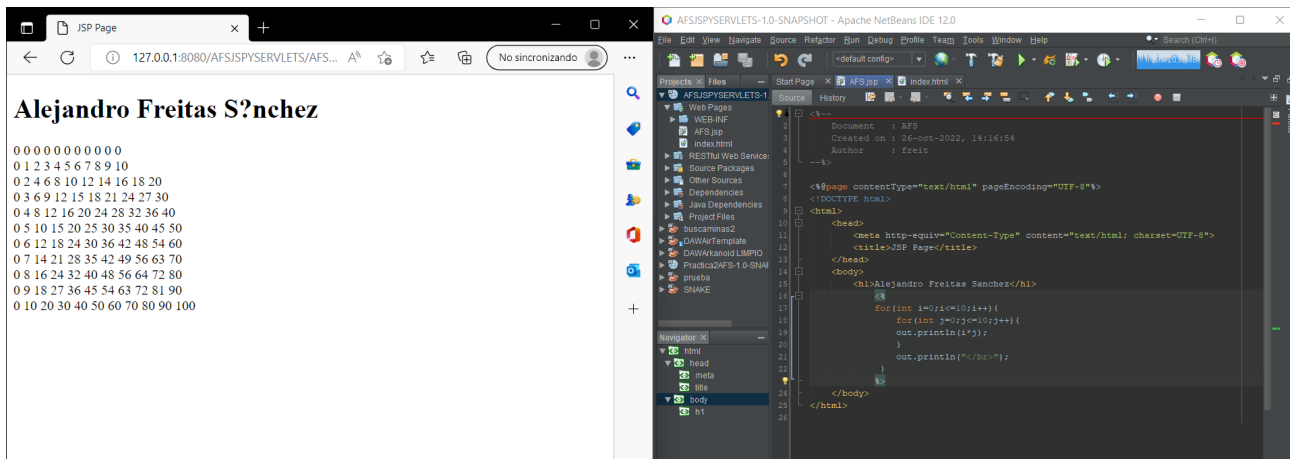
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Alejandro Freitas Sanchez</h1>
    <%
      for(int i=0;i<=10;i++){
        for(int j=0;j<=10;j++){
          out.println(i*j);
        }
        out.println("<br>");
      }
    %>
  </body>
</html>
```

4. Ejecutar en el contenedor la aplicación.



5. Mostrar la página y comparar el HTML con el contenido del JSP ¿Es diferente? ¿Qué ha sucedido?

Sí es diferente, el cliente no muestra el código que hay entre `<% %>`, a no ser que se imprima con `out.print`, pudiendo añadir etiquetas html dentro de el.



Cuestiones.

1. ¿Qué es el “Web Profile” de Jakarta? ¿Y el “MicroProfile”? Indicar al menos 3 elementos que se encuentren en el perfil web y no en el perfil micro.

El perfil web es un subconjunto específico de la plataforma Java EE completa, que se centra en los elementos de uso común.

El microProfile es la alternativa más ligera al uso de una implementación de perfil web, puede comenzar con una implementación de Java Servlet o JSP.

Elementos que se encuentran en el perfil web y no en el micro: API de transacciones de java, Bean Validation, Interfaz de proveedor de servicios de autenticación.

2. Buscar los servidores de aplicaciones que soportan y se han certificado Jakarta EE 9 “Full”.

FUJITSU Software Enterprise Application Platform, Payara server.

3. Buscar ahora los servidores de aplicaciones certificados para Jakarta EE 9 perfil web.

WildFly, Eclipse GassFish.

4. ¿A qué piensas que se debe la existencia de lo diferentes perfiles “Full”, “Web” “Core” y “Micro”? (Este último no es un perfil estándar).

A los distintos proyectos, cada proyecto no tiene los mismos requerimientos, por ejemplo una aplicación web que haga un alumno en una practica, va a tener menos requerimientos que una gran empresa, por lo que utilizará una versión más simple.

5. En la siguiente lista se detallan los elementos de Jakarta WebProfile 10:

Buscar información y explicar las función de los siguientes componentes:

- Jakarta Server Faces.

Es una especificación de Java para construir interfaces de usuario basadas en componentes para aplicaciones web y se formalizó como un estándar a través del Java Community Process que forma parte de Java Platform, Enterprise Edition.

- Jakarta WebSocket.

Es un componente que especifica la API que los desarrolladores pueden usar para integrar webSockets en sus aplicaciones, tanto en el lado servidor como cliente.

- Jakarta Persistence.

Es un componente que permite la gestión de datos en aplicaciones web.

- Jakarta Servlet.

Es un componente que permite que los servlets implementen contenedores web para alojar aplicaciones en servidores web.