

INTRO A JAVA WEB

Sin frameworks

Charly Cimino

Intro a Java Web (Sin frameworks)

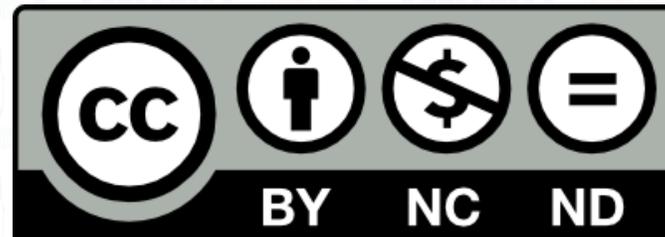
Charly Cimino

Este documento se encuentra bajo Licencia Creative Commons 4.0 Internacional (CC BY-NC-ND 4.0). Usted es libre para:

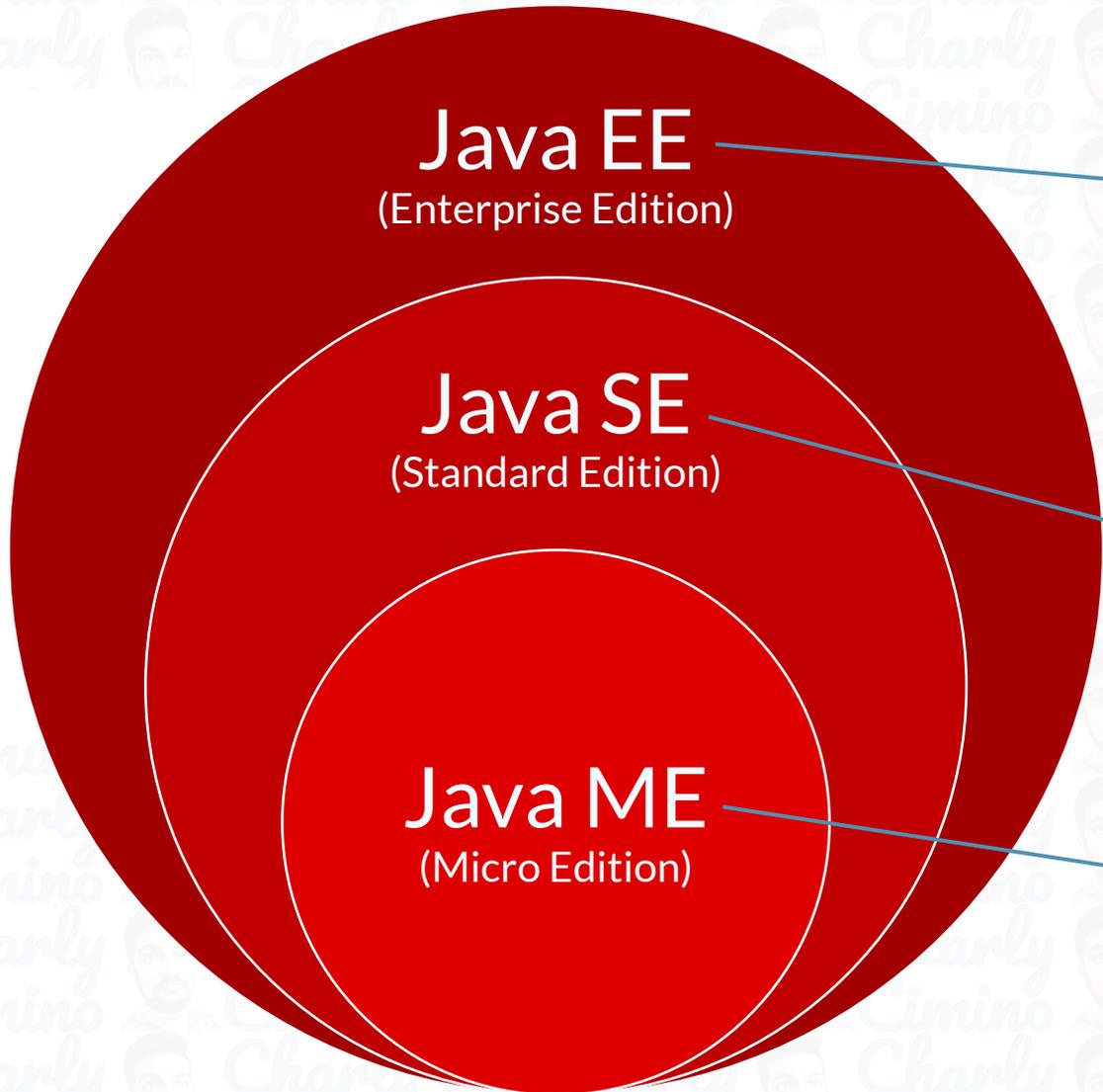
- **Compartir** — copiar y redistribuir el material en cualquier medio o formato.

Bajo los siguientes términos:

- **Atribución** — Usted debe darle crédito a esta obra de manera adecuada, proporcionando un enlace a la licencia, e indicando si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo del licenciante.
- **No Comercial** — Usted no puede hacer uso del material con fines comerciales.
- **Sin Derivar** — Si usted mezcla, transforma o crea nuevo material a partir de esta obra, usted no podrá distribuir el material modificado.



Plataformas de Java



Java EE
(Enterprise Edition)

Conjunto de paquetes, clases e interfaces para construir aplicaciones corporativas, ejecutables del lado del servidor.
A partir de 2018, lo gestiona la Fundación Eclipse y cambia de nombre a **Jakarta EE**.
No se incluye en el JDK (Java Development Kit)

Java SE
(Standard Edition)

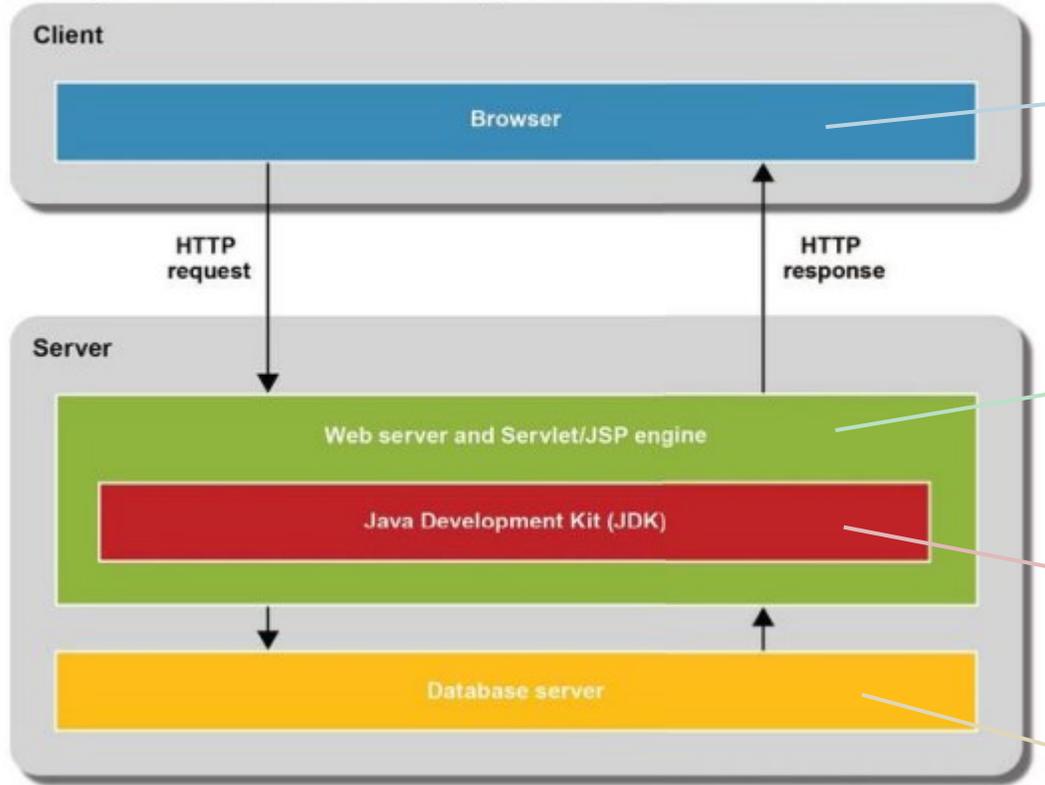
Contiene a Java ME y agrega paquetes, clases e interfaces elementales para construir aplicaciones de propósito general.

Java ME
(Micro Edition)

Contiene paquetes, clases e interfaces para construir aplicaciones ejecutables en dispositivos de prestaciones reducidas (Ej: un microondas).

Arquitectura de una app Web con Java

The components of a servlet/JSP application



Navegador web: software capaz de renderizar contenido web (por ejemplo, una página HTML)
Populares: Google Chrome, Brave, Firefox, Edge, Safari, Opera, etc.

Servidor web: software capaz de enviar respuestas HTTP a las peticiones de los clientes.
Motor de Servlet/JSP: Software exclusivo de la plataforma Java para generar respuestas dinámicas.

Incluye las APIs de Java SE, la VM (Virtual Machine), entre otras cosas necesarias para desarrollar y ejecutar aplicaciones de Java.

Servidor de base de datos: software capaz de proporcionar servicios a las aplicaciones para guardar y recuperar datos

Murach, J. and Urban, M. (2014) 'An Introduction to web programming with Java', in *Murach's Java Servlets and JSP: Training & reference*. Fresno, CA: Mike Murach & Associates, Inc., p. 15.

Aplicaciones web con Java

Para construir aplicaciones web con Java, se utilizarán las siguientes herramientas



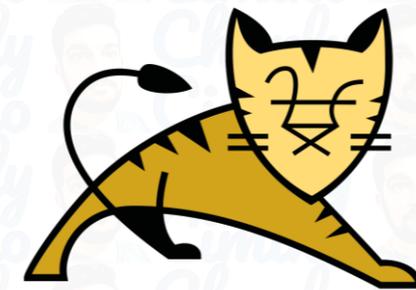
JDK 17

Por ser una versión LTS
(Long Time Support)



NetBeans 17

Preferiblemente



Apache Tomcat

Versión 10

Instrucciones de instalación

<https://youtu.be/2Et13pH2484>

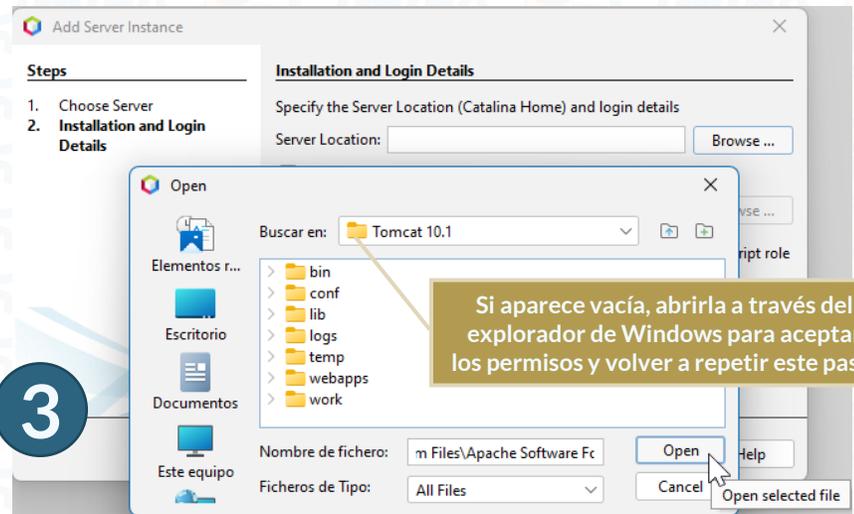
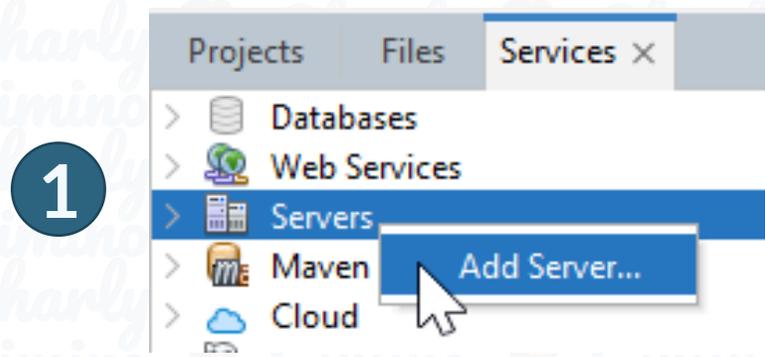
[Descargar de aquí](#)

Apache Tomcat es uno de los entornos más populares para el desarrollo web con Java. Incluye un servidor web llamado **Coyote** y un motor de Servlet/JSP llamado **Catalina**.

Agregar Tomcat a NetBeans

Ir a la pestaña **Services**. A continuación, hacer click derecho en **Servers** y seleccionar **Add Server...**

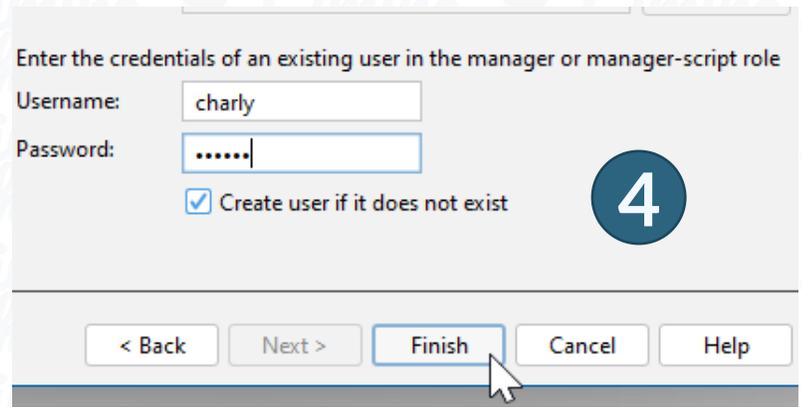
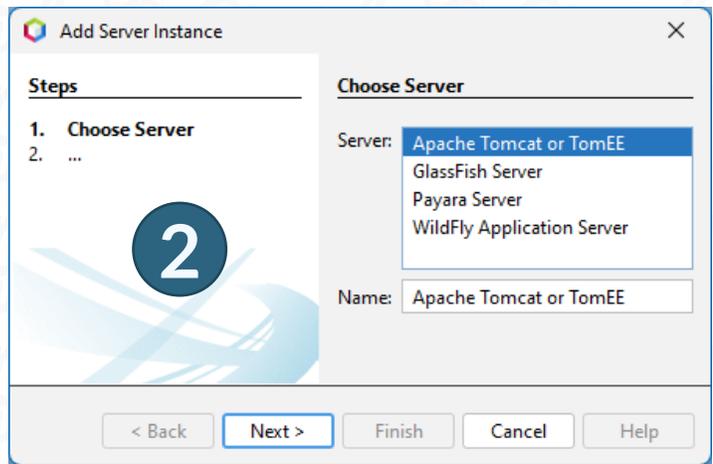
(Si no se ve la pestaña **Services**, entonces ir al menú **Window** y elegir **Services**)



Si aparece vacía, abrirla a través del explorador de Windows para aceptar los permisos y volver a repetir este paso.

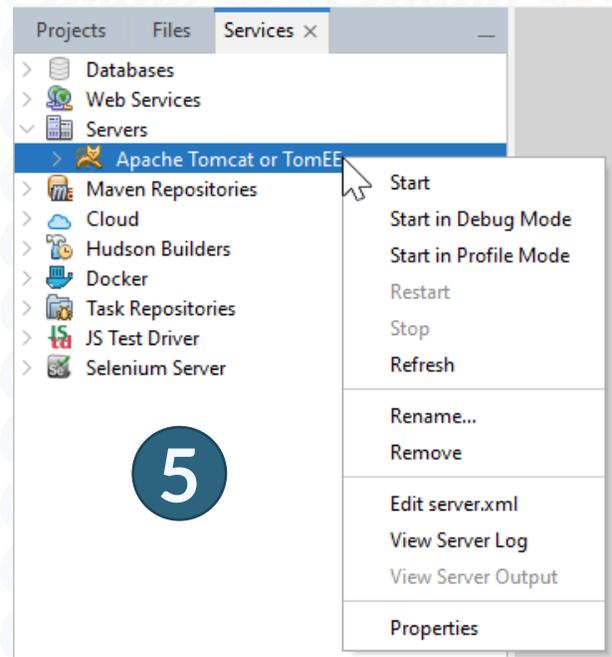
Click en **Browse...** para buscar la carpeta de instalación de Tomcat
Por defecto: C:\Program Files\Apache Software Foundation\Tomcat 10.1

Elegir **Apache Tomcat or TomEE** y luego click en **Next >**



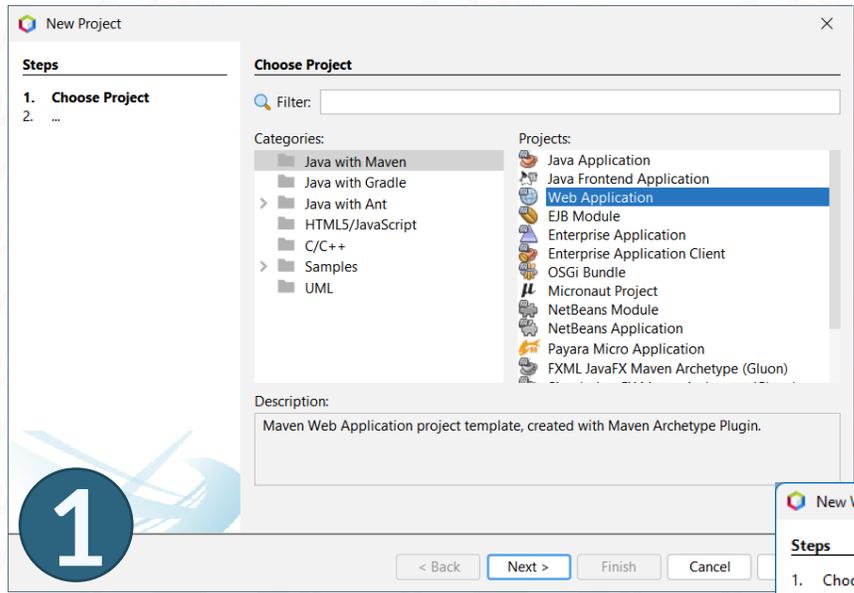
Colocar las mismas credenciales que en la instalación de Tomcat (u otras, si deseas crear un usuario nuevo como dice el checkbox)

Ya podemos administrar el servidor Tomcat desde Netbeans haciendo click derecho sobre él, en la pestaña **Services**.

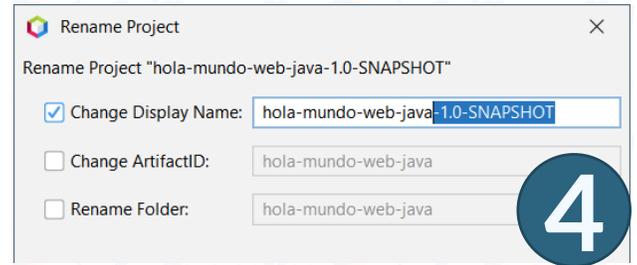
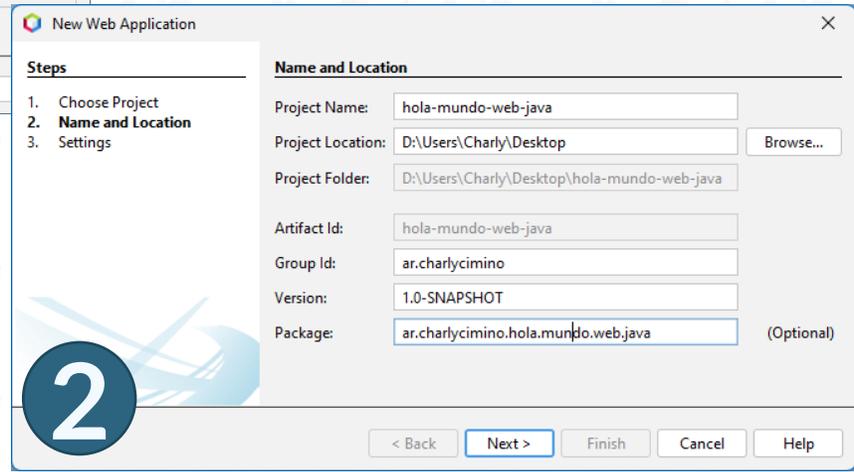
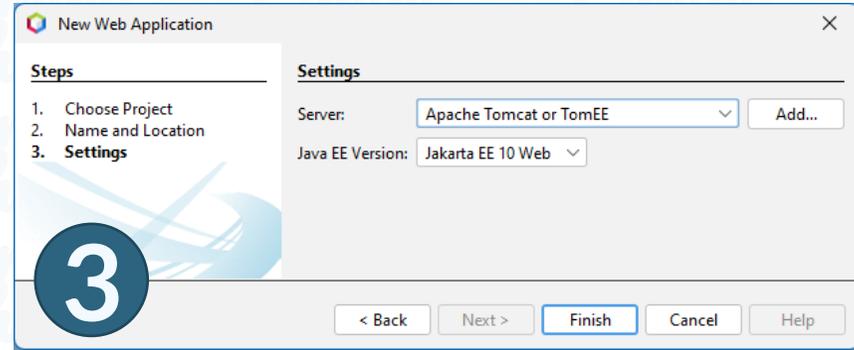


Crear app web en NetBeans

Ir al menú **File** y elegir **New Project**. A continuación, seleccionar **Web Application**, de la categoría **Java With Maven** *



Elegir **Apache Tomcat** (debe haberse agregado a Netbeans previamente) y la versión 10 de **Jakarta EE Web**.

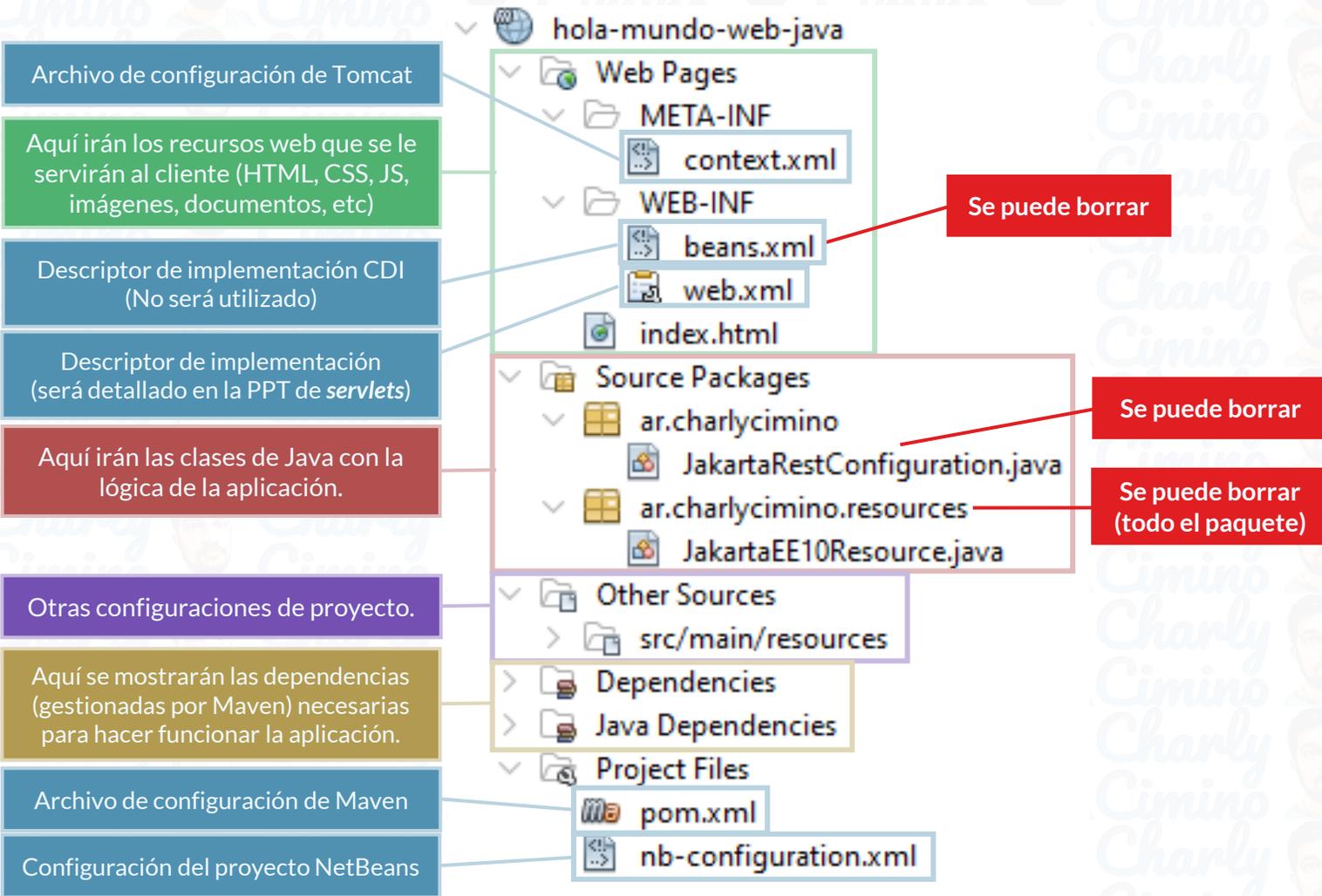


* **Maven es un gestor de dependencias para Java que simplifica la manipulación de librerías de terceros.**

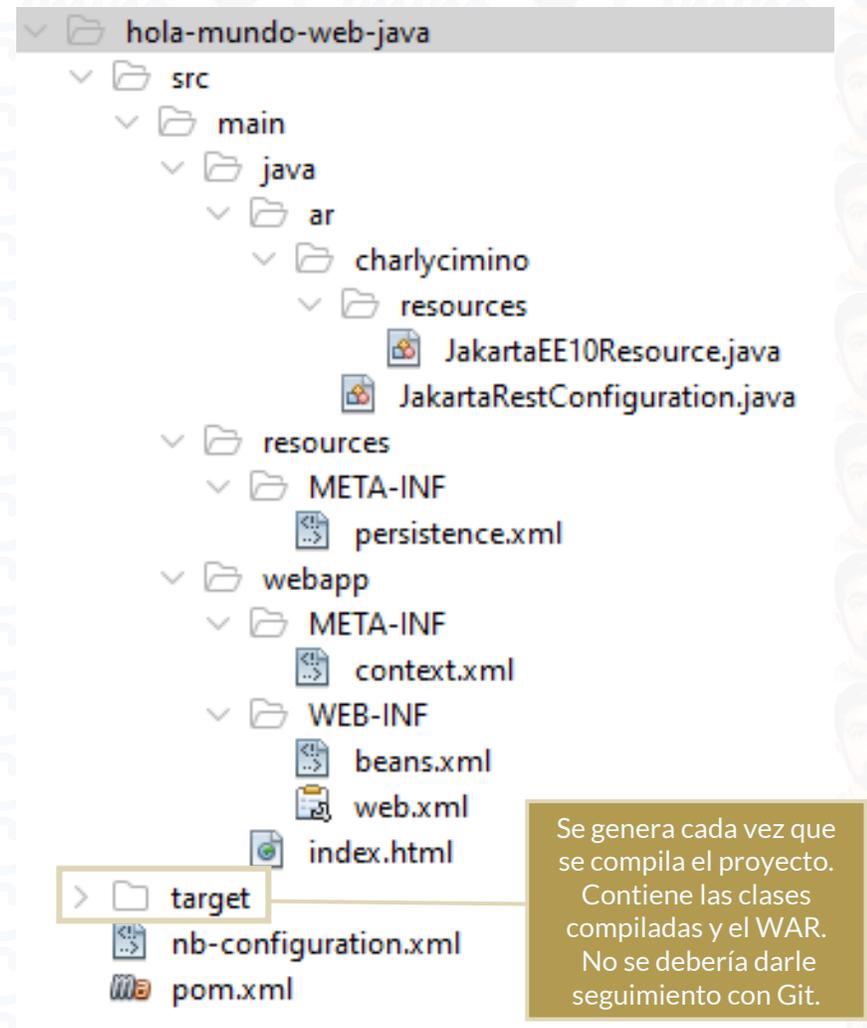
Hacer click derecho sobre el proyecto y elegir la opción **Rename** para cambiar el nombre para mostrar. (borrar el **-1.0-SNAPSHOT**).

Completar los campos prestando atención al nombre del proyecto y su localización en el disco. Los campos **Group ID**, **Version** y **Package** no son tan relevantes cuando recién se empieza.

Proyecto Java Web con Maven y Tomcat



Disposición lógica
(Como se administra en NetBeans)



Disposición real
(Como se guarda en el disco)

Correr el proyecto web

2

Probablemente la primera vez se obtenga este error:

```
Failed to execute goal org.apache.maven.plugins:maven-war-plugin:2.3:war (default-war) on project hola-mundo-web-java: Execution default-war of goal org.apache.maven.plugins:maven-war-plugin:2.3:war failed: Unable to load the mojo 'war' in the plugin 'org.apache.maven.plugins:maven-war-plugin:2.3' due to an API incompatibility: org.codehaus.plexus.component.repository.exception.ComponentLookupException: null
```

3

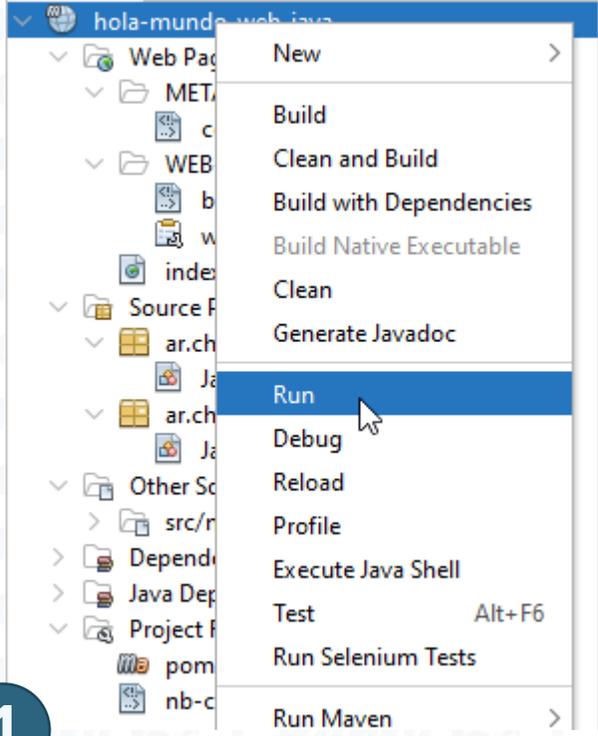
Abrir el archivo pom.xml y ubicar la siguiente declaración:

```
pom.xml
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-war-plugin</artifactId>
  <version>2.3</version>
  <configuration>
    <failOnMissingWebXml>>false</failOnMissingWebXml>
  </configuration>
</plugin>
```

Cambiar por 3.3.2

1

Hacer click derecho sobre el proyecto y elegir la opción Run



4

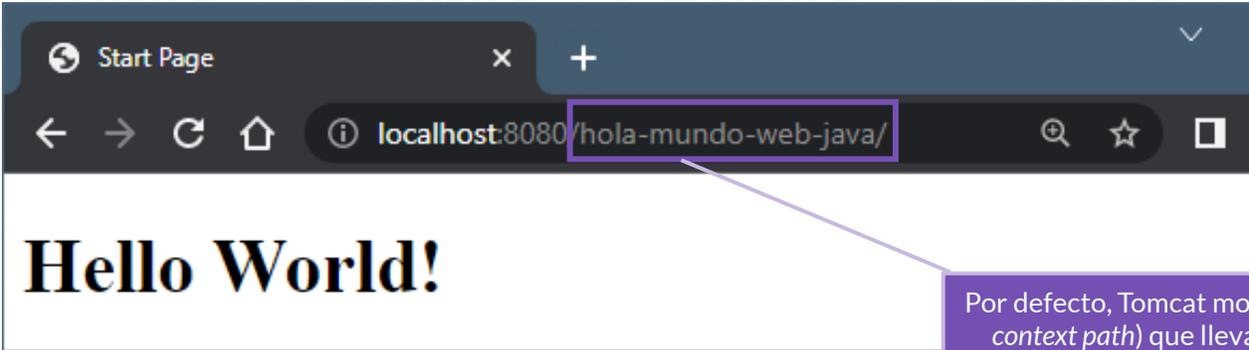
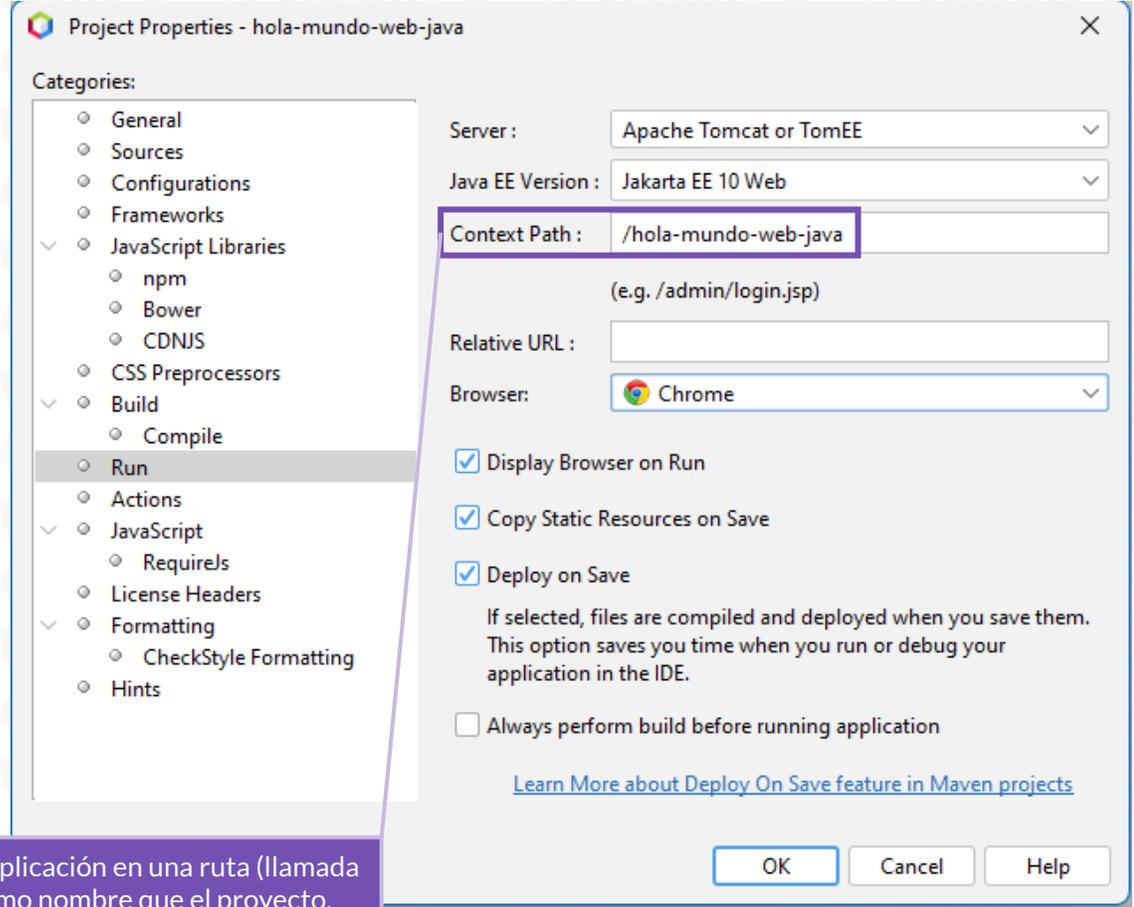
Guardar el archivo (Ctrl + S) y volver a correr el proyecto

Proyecto en ejecución

```
index.html
<!DOCTYPE html>
<html>
  <head>
    <title>Start Page</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

Se abrirá el browser y hará una petición a la ruta raíz de la aplicación, que servirá el contenido de la página de bienvenida (index.html)

Si no lo hace automáticamente, se puede intentar realizar manualmente



Por defecto, Tomcat monta la aplicación en una ruta (llamada *context path*) que lleva el mismo nombre que el proyecto. Para cambiarla, hacer click derecho sobre el proyecto, seleccionar **Properties** e ir a la categoría **Run**.

Comprendiendo la URL

`http://localhost:8080/ho1a-mundo-web-java`

El tipo de protocolo utilizado.
(Otros pueden ser `file:///`, `ftp://`, `mysql://`, etc)

El número de puerto que utiliza Tomcat para enviar respuestas. Por ejemplo, MySQL Server utiliza el puerto 3306.

La ruta de contexto (*context path*), que hace referencia a nuestra app web. Necesaria porque podría haber otras apps corriendo en el mismo server.

`localhost` es un nombre reservado, que representa la conexión hacia nosotros mismos (*loopback*), a través de la IP reservada para tal fin (`127.0.0.1`).

La siguiente URL produce el mismo resultado:
`http://127.0.0.1:8080/ho1a-mundo-web-java`



Cliente / Servidor

Nuestra computadora hace de cliente y servidor a la vez.

Para más detalle, consultá mi ppt: [“El protocolo HTTP”](#)

Ciente-Servidor con dos dispositivos

Abri la consola de Windows (escribi `cmd` en el cuadro de búsqueda)

```

Símbolo del sistema
Microsoft Windows [Versión 10.0.22621.2134]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Charly>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

Sufijo DNS específico para la conexión. . . :
Dirección IPv6 . . . . . : 2800:810:440:1b9c::1000
Dirección IPv6 . . . . . : 2800:810:440:1b9c:ab38:4cb4:8a30:20a1
Dirección IPv6 temporal. . . . . : 2800:810:440:1b9c:2d3f:2c7b:a728:3170
Vínculo: dirección IPv6 local. . . : fe80::f00d:41d7:694c:c75e%12
Dirección IPv4. . . . . : 192.168.0.82
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . : fe80::2a9e:fcff:fe42:9549%12
                                         192.168.0.1

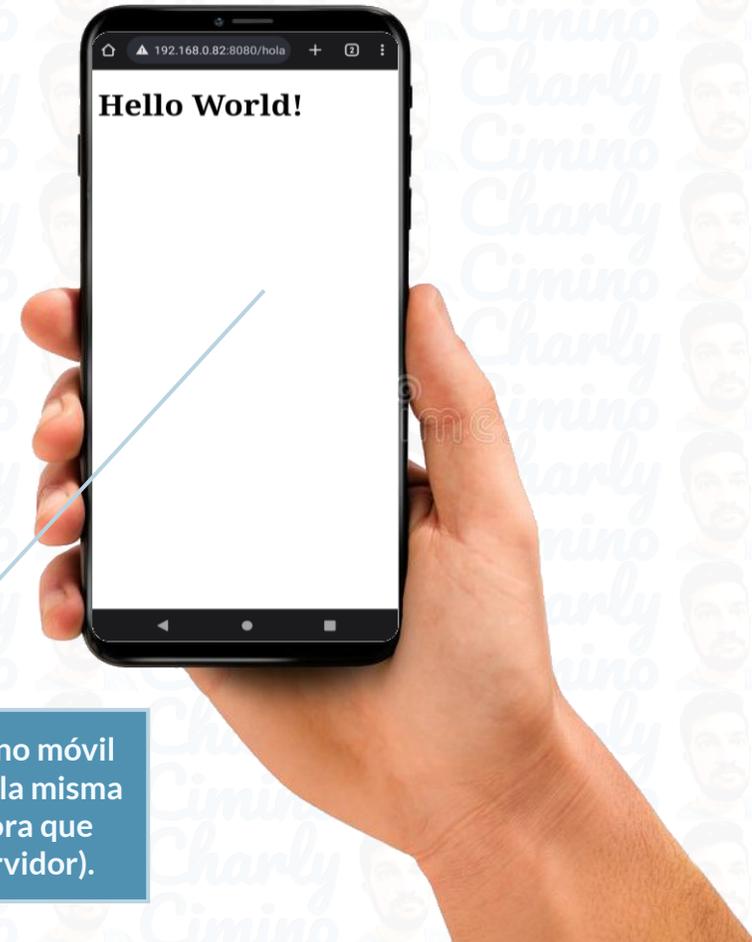
```

Escribi el comando `ipconfig` y presiona Enter.
(En Linux, es `ifconfig`)

Esta es la IP de tu computadora en tu red de área local (LAN),

Abri un *browser* en otro dispositivo conectado a la misma red de tu hogar y reemplazá `localhost` por la IP de tu servidor.

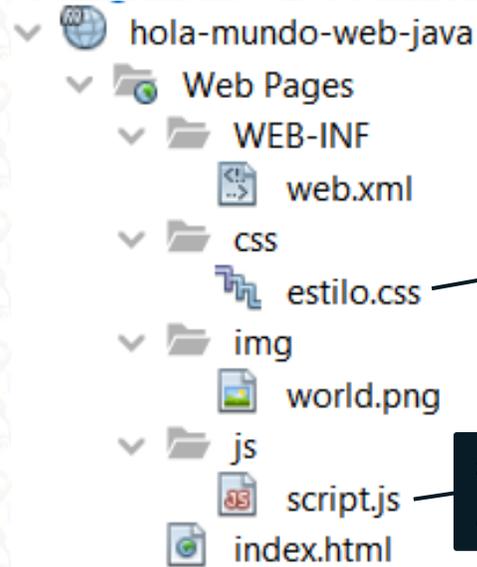
`http://tu.ip.local:8080/hola-mundo-web-java`



Resultado en un teléfono móvil (el cliente) conectado a la misma red que la computadora que ejecuta Tomcat (el servidor).

Servir contenido estático

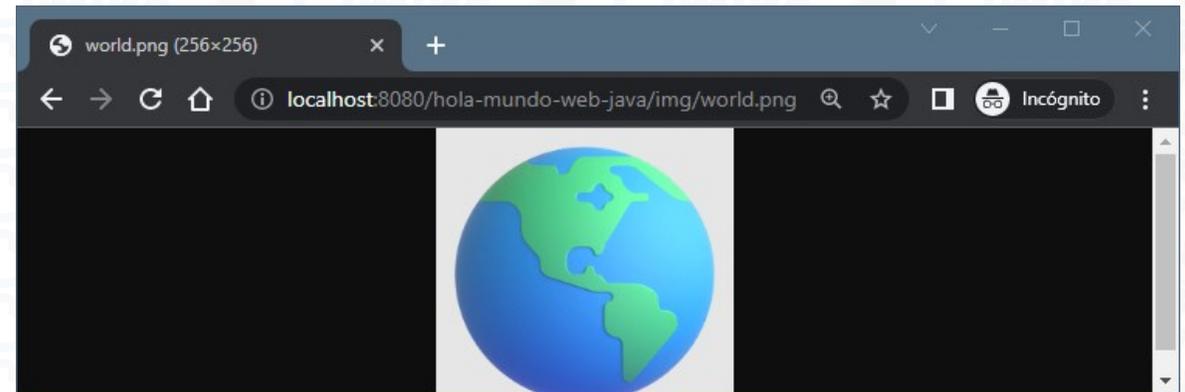
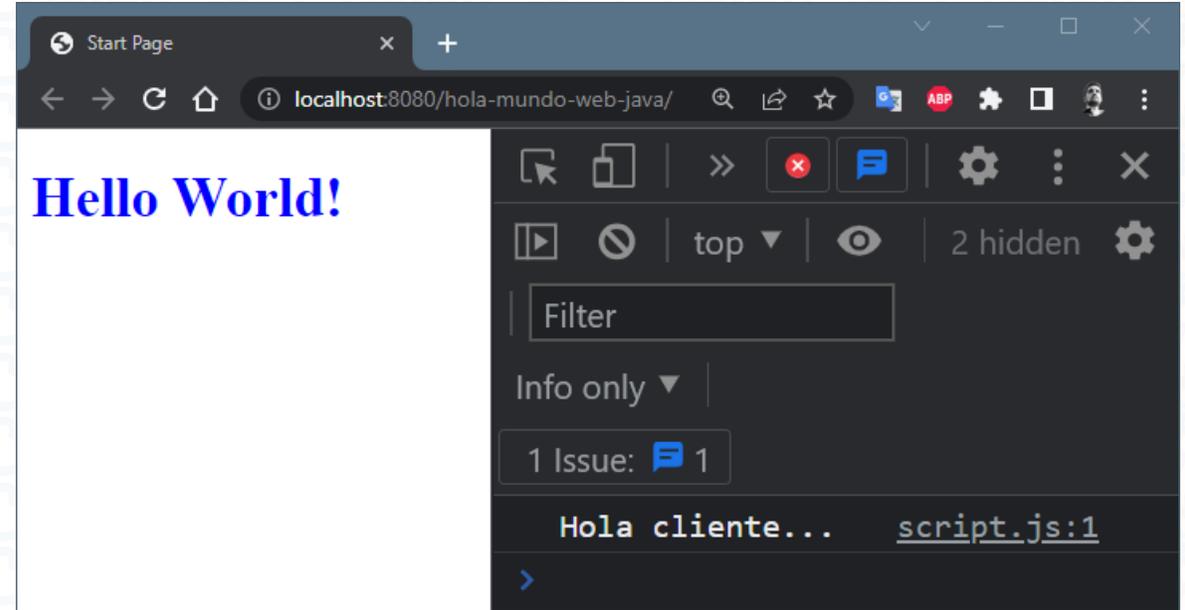
Todos los recursos colocados en la carpeta lógica **Web Pages** (ubicación real: **/webapp**) podrían ser accedidos por el cliente, conociendo la URL, excepto el contenido de las carpetas **WEB-INF** y **META-INF** que son privadas.



```
estilo.css
h1 {
  color: blue;
}
```

```
script.js
console.log("Hola cliente...");
```

```
index.html
<!DOCTYPE html>
<html>
  <head>
    <title>Start Page</title>
    <link rel="stylesheet" href="css/estilo.css"/>
  </head>
  <body>
    <h1>Hello World!</h1>
    <script src="js/script.js"></script>
  </body>
</html>
```



Servir contenido dinámico

Ver PPT de...

Java Web: Servlets

(Click para acceder)

FIN DE LA PRESENTACIÓN

Encontrá más como estas en mi [sitio web](#).