



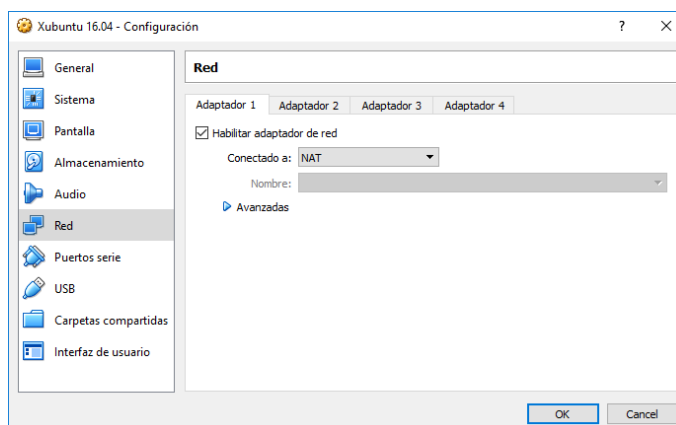
## Boletín de ejercicios Tema 1

---

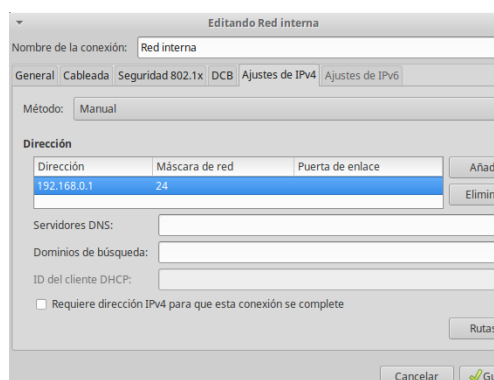
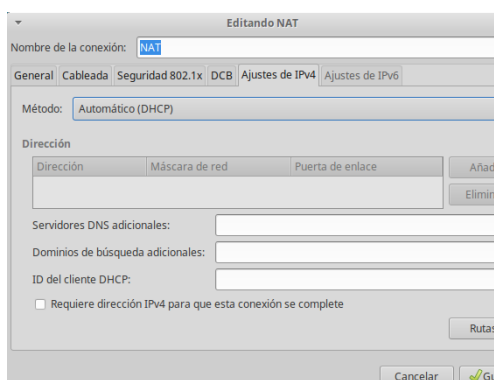
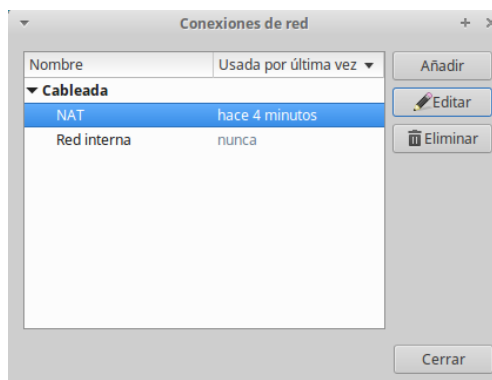
### 1. Instalación de Tomcat en Xubuntu 20.04 y comprobación de su funcionamiento

#### SOLUCIÓN

En primer lugar, configuraremos el modo de red en VirtualBox para la máquina servidor Xubuntu 20.04 como NAT, para poder tener conexión a Internet y hacer la instalación. Para eso, presionaremos con el botón derecho sobre esa máquina en VirtualBox y escogeremos Configuración. A continuación, escogeremos las siguientes opciones:



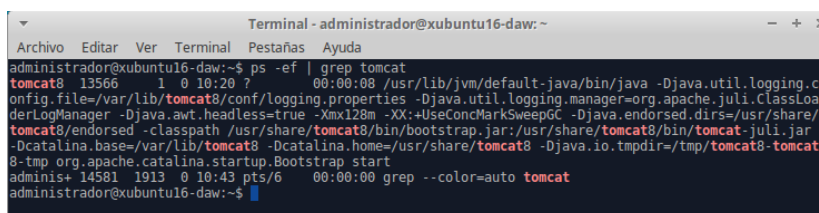
Lanzamos la máquina Xubuntu 20.04 servidor y crearemos dos configuraciones de red diferentes para poder emplear la máquina en modo NAT y Red interna y conmutar fácilmente entre ellos:



Para esta vez, escogemos la configuración NAT. Abrimos una terminal y comenzamos la instalación:

```
sudo apt-get install openjdk-8-jdk tomcat8 tomcat8-examples
```

Cuando acabe, comprobamos que el proceso esté activo:

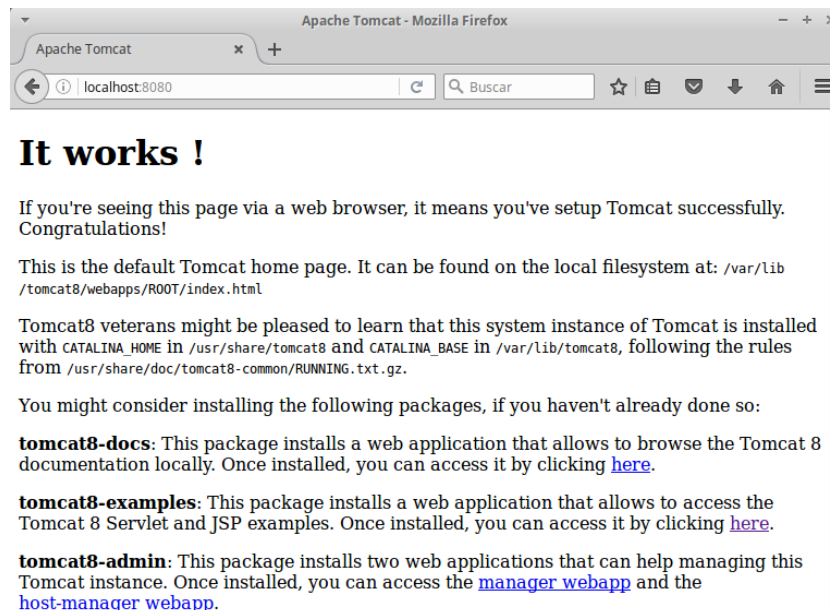


Y escuchando en el puerto 8080:



```
Terminal - administrador@xubuntu16-daw: ~
Archivo Editar Ver Terminal Pestañas Ayuda
administrador@xubuntu16-daw:~$ netstat -ltn
Conexiones activas de Internet (solo servidores)
Proto Recib Enviad Dirección local Dirección remota Estado
tcp 0 0 127.0.1.1:53 0.0.0.0:* ESCUCHAR
tcp6 0 0 :::443 :::* ESCUCHAR
tcp6 0 0 192.168.0.1:8085 :::* ESCUCHAR
tcp6 0 0 :::8080 :::* ESCUCHAR
tcp6 0 0 :::80 :::* ESCUCHAR
tcp6 0 0 :::21 :::* ESCUCHAR
administrador@xubuntu16-daw:~$
```

Abrimos el navegador y tecleamos **http://localhost:8080:**



Cambiamos la configuración de red a “Red interna” tanto en VirtualBox como en la interfaz del gestor de red.

Lanzamos la máquina Xubuntu 18.04 cliente, configurándola en Red interna como se hizo para el servidor, abrimos un navegador y tecleamos `http://192.168.0.1:8080`, que es la IP de la máquina servidor y obtenemos la misma página web.

Escogemos Servlets examples:

### Apache Tomcat Examples

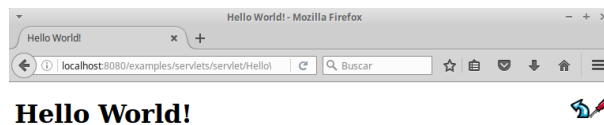
- [Servlets examples](#)
- [JSP Examples](#)
- [WebSocket Examples](#)

Y ejecutamos uno de ellos:



Hello World	<a href="#">Execute</a>	<a href="#">Source</a>
Request Info	<a href="#">Execute</a>	<a href="#">Source</a>
Request Headers	<a href="#">Execute</a>	<a href="#">Source</a>
Request Parameters	<a href="#">Execute</a>	<a href="#">Source</a>
Cookies	<a href="#">Execute</a>	<a href="#">Source</a>
Sessions	<a href="#">Execute</a>	<a href="#">Source</a>

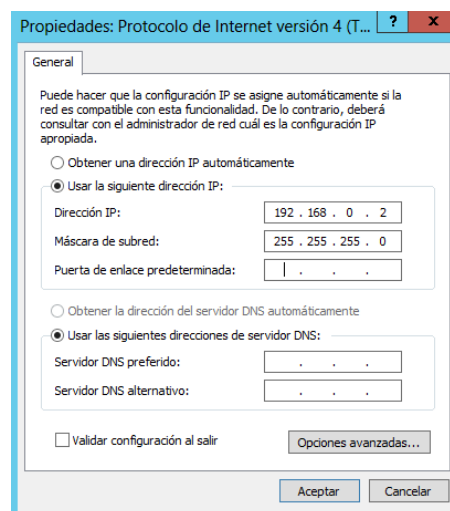
Obteniendo:



## 2. Instalación de Tomcat en Windows 10 y comprobación de su funcionamiento.

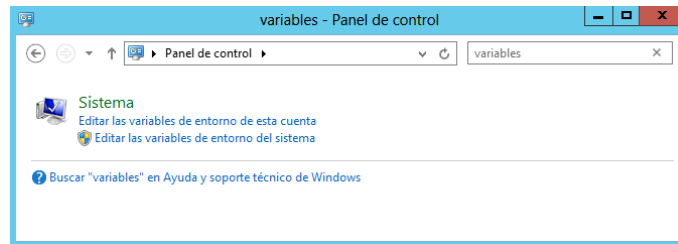
### SOLUCIÓN

Después de escoger en la configuración de VirtualBox la opción de red “Red interna” para Windows 10, arrancaremos la máquina y configuraremos las opciones de red con las siguientes opciones:

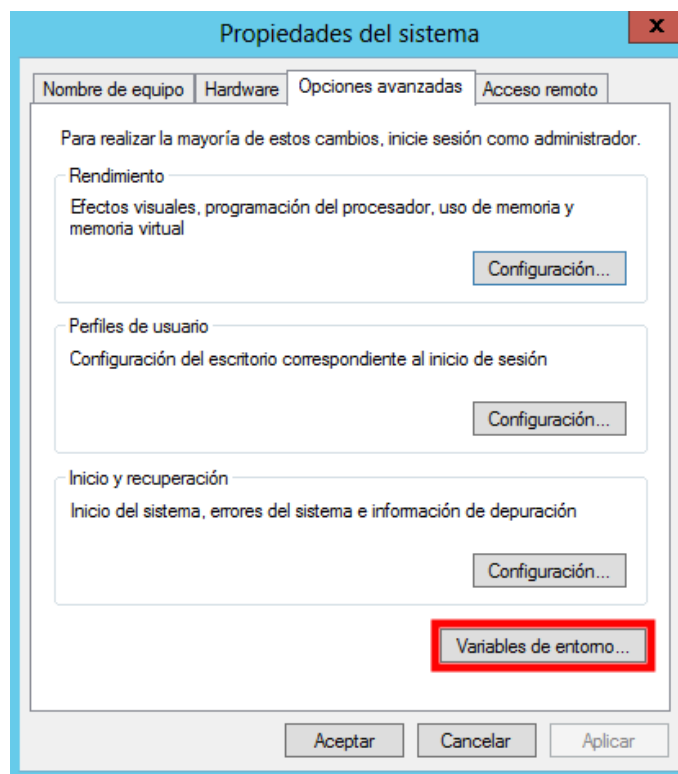


Descargamos el Java Development Kit y lo instalamos.

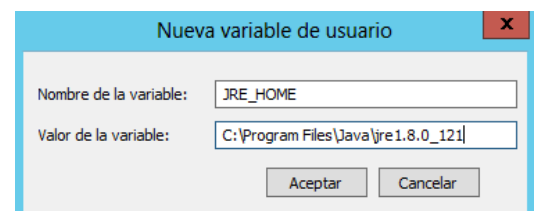
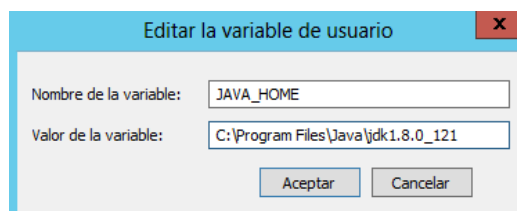
Editamos las variables de entorno del sistema (desde Panel de control, buscamos variables):



Hacemos clic en Variables de entorno...



Y añadimos estas dos:

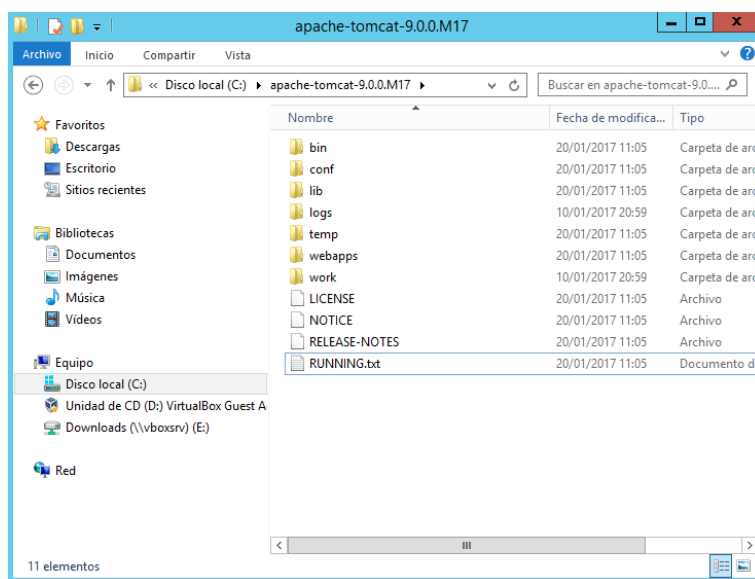




Reiniciamos el servidor para que los cambios tengan efecto.

A continuación, descargamos los archivos precisos para la instalación de Apache Tomcat desde la máquina real de la dirección.

Una vez reiniciada, descomprimos el fichero apache-tomcat-9.0.0.M17.zip, extrayendo la carpeta apache-tomcat-9.0.0.M17 que se encuentra en su interior en C:\



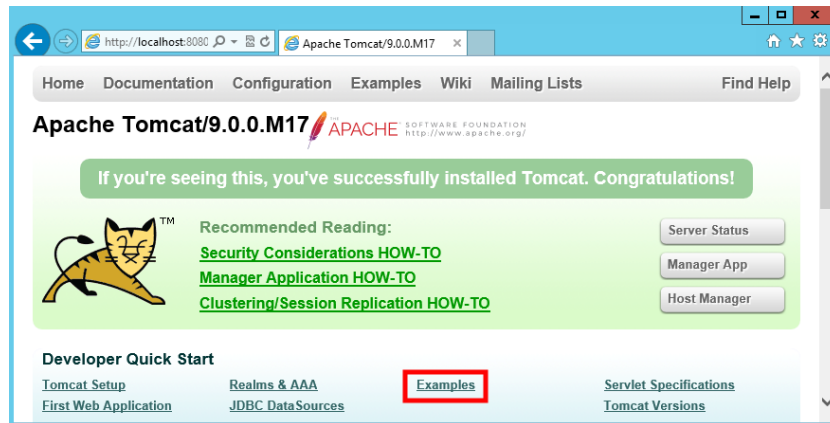
Abrimos una consola de comandos e instalamos Tomcat 9:

```
Administrador: Símbolo del sistema - tomcat9.exe -k install
Microsoft Windows [Versión 6.2.9200]
(c) 2012 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Administrador>cd c:\apache-tomcat-9.0.0.M17\bin
c:\apache-tomcat-9.0.0.M17\bin>tomcat9.exe -k install
20-Jan-2017 11:32:35.016 INFORMACIÓN [main] org.apache.catalina.startup.VersionL
oggerListener.log Server version: Apache Tomcat/9.0.0.M17
20-Jan-2017 11:32:35.016 INFORMACIÓN [main] org.apache.catalina.startup.VersionL
oggerListener.log Server built: Jan 10 2017 20:59:20 UTC
20-Jan-2017 11:32:35.016 INFORMACIÓN [main] org.apache.catalina.startup.VersionL
oggerListener.log Server number: 9.0.0.0
20-Jan-2017 11:32:35.016 INFORMACIÓN [main] org.apache.catalina.startup.VersionL
oggerListener.log OS Name: Windows Server 2012
20-Jan-2017 11:32:35.016 INFORMACIÓN [main] org.apache.catalina.startup.VersionL
oggerListener.log OS Version: 6.2
20-Jan-2017 11:32:35.016 INFORMACIÓN [main] org.apache.catalina.startup.VersionL
oggerListener.log Architecture: amd64
20-Jan-2017 11:32:35.016 INFORMACIÓN [main] org.apache.catalina.startup.VersionL
oggerListener.log Java Home: C:\Program Files\Java\jre1.8.0_121
20-Jan-2017 11:32:35.016 INFORMACIÓN [main] org.apache.catalina.startup.VersionL
oggerListener.log JVM Version: 1.8.0_121-b13
20-Jan-2017 11:32:35.016 INFORMACIÓN [main] org.apache.catalina.startup.VersionL
oggerListener.log JVM Vendor: Oracle Corporation
20-Jan-2017 11:32:35.033 INFORMACIÓN [main] org.apache.catalina.startup.VersionL
```

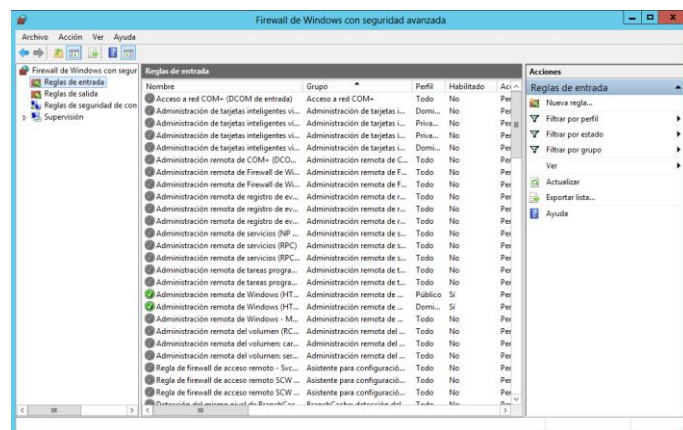


Lanzamos Tomcat 9 ejecutando C:\apache-tomcat-9.0.0.M17\bin\startup.bat y comprobamos que funciona correctamente tecleando en el navegador de Internet local <http://localhost:8080>:



Podemos comprobar que los Servlets se ejecutan correctamente haciendo clic en Examples y procediendo al igual que en la tarea anterior.

Para conseguir acceder desde la máquina cliente, tenemos que configurar el cortafuegos de Windows para que acepte conexiones entrantes en el puerto 8080. Para eso, crearemos una nueva regla entrante:





**Asistente para nueva regla de entrada**

**Tipo de regla**  
Seleccione el tipo de regla de firewall que desea crear.

**Pasos:**

- Tipo de regla
- Protocolo y puertos
- Acción
- Perfil
- Nombre

¿Qué tipo de regla desea crear?

☐ Programa  
Regla que controla las conexiones de un programa.

☒ Puerto  
Regla que controla las conexiones de un puerto TCP o UDP.

☐ Predefinida:  
Acceso a red COM+  
Regla que controla las conexiones de una experiencia con Windows.

☐ Personalizada  
Regla personalizada.

< Atrás    Siguiendo >    Cancelar

**Asistente para nueva regla de entrada**

**Protocolo y puertos**  
Especifique los puertos y protocolos a los que se aplica esta regla.

**Pasos:**

- Tipo de regla
- Protocolo y puertos
- Acción
- Perfil
- Nombre

¿Se aplica esta regla a TCP o UDP?

☒ TCP

☐ UDP

¿Se aplica esta regla a todos los puertos locales o a unos puertos locales específicos?

☐ Todos los puertos locales

☒ Puertos locales específicos: 8080  
Ejemplo: 80, 443, 5000-5010

< Atrás    Siguiendo >    Cancelar

**Asistente para nueva regla de entrada**

**Acción**  
Especifique la acción que debe llevarse a cabo cuando una conexión coincide con las condiciones especificadas en la regla.

**Pasos:**

- Tipo de regla
- Protocolo y puertos
- Acción
- Perfil
- Nombre

¿Qué medida debe tomarse si una conexión coincide con las condiciones especificadas?

☒ Permitir la conexión  
Esto incluye las conexiones protegidas mediante IPsec y las que no lo están.

☐ Permitir la conexión si es segura  
Esto incluye solamente las conexiones autenticadas mediante IPsec. Éstas se protegerán mediante la configuración de reglas y propiedades de IPsec del nodo Regla de seguridad de conexión.

Personalizar...

☐ Bloquear la conexión

< Atrás    Siguiendo >    Cancelar

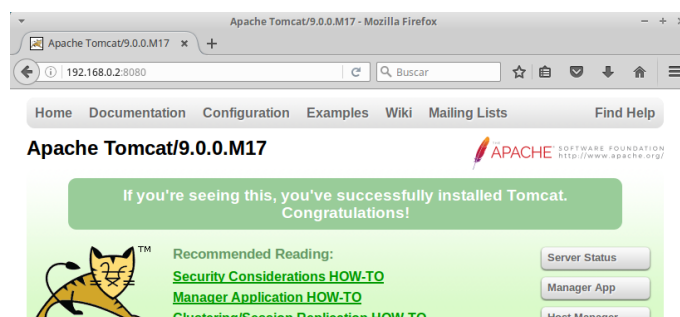




The screenshot shows the 'Perfil' step of the 'Asistente para nueva regla de entrada' wizard. On the left, a 'Pasos:' (Steps) list includes 'Tipo de regla', 'Protocolo y puertos', 'Acción', 'Perfil', and 'Nombre'. The 'Perfil' step is selected. The main area asks '¿Cuándo se aplica esta regla?' (When does this rule apply?) and lists three options with checkboxes: 'Dominio' (checked), 'Privado' (checked), and 'Público' (checked). Each option has a description: 'Dominio' applies when connected to a corporate domain; 'Privado' applies when connected to a private network like home or work; 'Público' applies when connected to a public network. Navigation buttons at the bottom are '< Atrás', 'Siguiendo >', and 'Cancelar'.

The screenshot shows the 'Nombre' step of the 'Asistente para nueva regla de entrada' wizard. The 'Pasos:' list on the left has 'Nombre' selected. The main area asks 'Especifique el nombre y la descripción de esta regla.' (Specify the name and description of this rule.) It contains a 'Nombre:' text field with 'Apache' entered and a 'Descripción (opcional):' (optional description) text area. Navigation buttons at the bottom are '< Atrás', 'Finalizar', and 'Cancelar'.

Después, comprobamos que podemos acceder desde la máquina cliente, tecleando <http://192.168.0.2:8080>:





**3. Dispones de una máquina que cuenta con el sistema operativo Xubuntu 20.04, esta máquina tiene el entorno de red configurado y, además, dispones de conexión a Internet. Además, estás trabajando con la cuenta del usuario root. Indica cada uno de los pasos, y comandos implicados en ellos, para conseguir hacer lo siguiente:**

- a. Instalar el servidor web Apache desde terminal
- b. Comprobar que está funcionando el servidor Apache desde terminal
- c. Comprobar que está funcionando el servidor Apache desde el navegador
- d. Cambiar el puerto por el cual está escuchando Apache pasándolo al puerto 82

### **SOLUCIÓN**

- a. Accedemos a una pantalla de terminal y nos logamos como usuario root tecleando `sudo su` y poniendo la contraseña de root que tengamos.

Con la siguiente orden lo instalamos: `apt-get install apache2`

- b. Para comprobar el funcionamiento de apache desde terminal, estando como root teclearemos: `/etc/init.d/apache2 status`

Este comando nos informará si se está ejecutando y el número de procesos que tiene.

- c. Abrimos un navegador y en su barra de direcciones tecleamos `localhost`, tras lo que nos aparecerá el mensaje "It works!"

- d. Para cambiar el puerto de escucha de Apache editamos el fichero **ports.conf** que se encuentra en **etc/apache2**, por lo que en un terminal teclearemos: `gedit /etc/apache2/ports.conf`

Cambiaremos las líneas referidas al puerto 80, que es el que deseamos cambiar, sustituyendo el puerto 80 por el 82:

1. `NameVirtualHost *:80`
2. `Listen 80`



Reiniciaremos el servidor tecleando: `/etc/init.d/apache2 restart`

Comprobaremos que si en un navegador tecleamos **localhost** nos saldrá una página de error, pero si tecleamos **localhost:82** nos aparecerá la página de "It Works!"

#### 4. Desplegar una aplicación en el servidor de aplicaciones

##### SOLUCIÓN

Código adjunto

#### 5. Creación de un servlet

##### SOLUCIÓN

Código adjunto

#### 6. Autenticación y autorización

##### SOLUCIÓN

- En la máquina servidor Xubuntu, crearemos los usuarios y rol en Tomcat, añadiendo al fichero `/var/lib/tomcat9/conf/tomcat-users.xml` las siguientes líneas:

```
<role rolename="usuarios-servlet"/>
<user username="usuario1" password="contrasenha" roles="usuarios-servlet"/>
<user username="usuario2" password="contrasenha" roles="usuarios-servlet"/>
```

- Reiniciamos el servidor Tomcat:



```
sudo service tomcat9 restart
```

- En la máquina donde hayamos instalado Netbeans, importaremos la carpeta del proyecto Servlet Formulario.
- Configuramos el Realm, para eso, dentro de este proyecto, creamos el fichero META-INF/context.xml, haciendo clic derecho sobre Web Content/META-INF y escogiendo New...→ File y escribimos las siguientes líneas:

```
<Context>  
  <Realm className="org.apache.catalina.realm.MemoryRealm"/>  
</Context>
```

- Protegemos la aplicación con el MemoryRealm, empleando autenticación BASIC, para lo cual editaremos el descriptor de despliegue de la aplicación web (web.xml) añadiendo los siguientes elementos como hijos del elemento raíz <web-app>.

```
<security-constraint>  
  <web-resource-collection>  
    <web-resource-name>Servlet Factorial</web-resource-name>  
    <url-pattern>/*</url-pattern>  
  </web-resource-collection>  
  <auth-constraint>  
    <role-name>usuarios-servlet</role-name>  
  </auth-constraint>  
</security-constraint>  
<login-config>  
  <auth-method>BASIC</auth-method>  
  <realm-name>Acceso al factorial</realm-name>  
</login-config>
```



- Eliminamos la aplicación del servidor Tomcat (si ya la habíamos desplegado anteriormente), creamos el fichero WAR y la volvemos a desplegar.
- Comprobamos que la configuración fue la adecuada, accediendo desde el navegador de la máquina cliente a `http://192.168.0.1:8080/Factorial`. Comprobamos que es necesario introducir el usuario y contraseña para acceder a la aplicación.

## 7. Despliegue de aplicaciones web asegurando las comunicaciones entre cliente y servidor

### SOLUCIÓN

- Crearemos un almacén de claves con un certificado SSL, utilizando el siguiente comando:

```
sudo keytool -genkey -alias tomcat -keyalg RSA -keystore /var/lib/tomcat9/claves
```

```
administrador@xubuntu16-daw:~$ sudo keytool -genkey -alias tomcat -keyalg RSA -keystore /var/lib/tomcat8/claves
[sudo] password for administrador:
Introduzca la contraseña del almacén de claves:
Volver a escribir la contraseña nueva:
¿Cuáles son su nombre y su apellido?
[Unknown]: Pepito Pérez González
¿Cuál es el nombre de su unidad de organización?
[Unknown]: Xunta de Galicia
¿Cuál es el nombre de su organización?
[Unknown]: Consellería de Educación
¿Cuál es el nombre de su ciudad o localidad?
[Unknown]: Vigo
¿Cuál es el nombre de su estado o provincia?
[Unknown]: Pontevedra
¿Cuál es el código de país de dos letras de la unidad?
[Unknown]: ES
¿Es correcto CN=Pepito Pérez González, OU=Xunta de Galicia, O=Consellería de Educación, L=Vigo, ST=Pontevedra, C=ES?
[no]: sí
Introduzca la contraseña de clave para <tomcat>
(INTRO sí es la misma contraseña que la del almacén de claves):
```



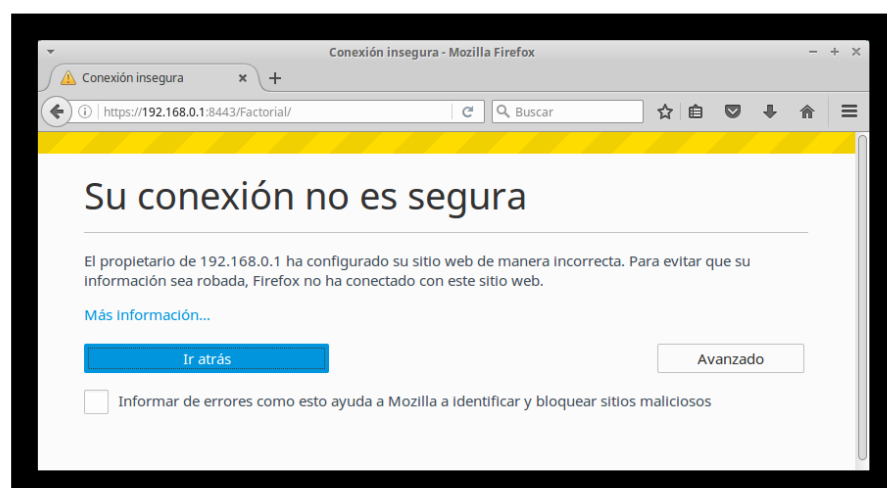
- Configuramos un conector SSL en Tomcat, editando el fichero `/var/lib/tomcat8/conf/server.xml` y añadiendo las siguientes líneas dentro del elemento raíz `<Server>`:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="/var/lib/tomcat8/almacen_claves"
keystorePass="tomcat"
keyAlias="tomcat" keyPass="tomcat"/>
```

- Reiniciamos el servidor Tomcat:

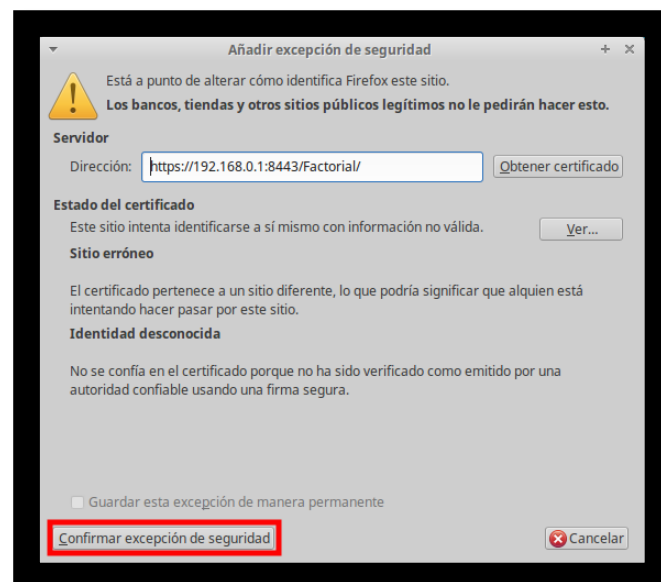
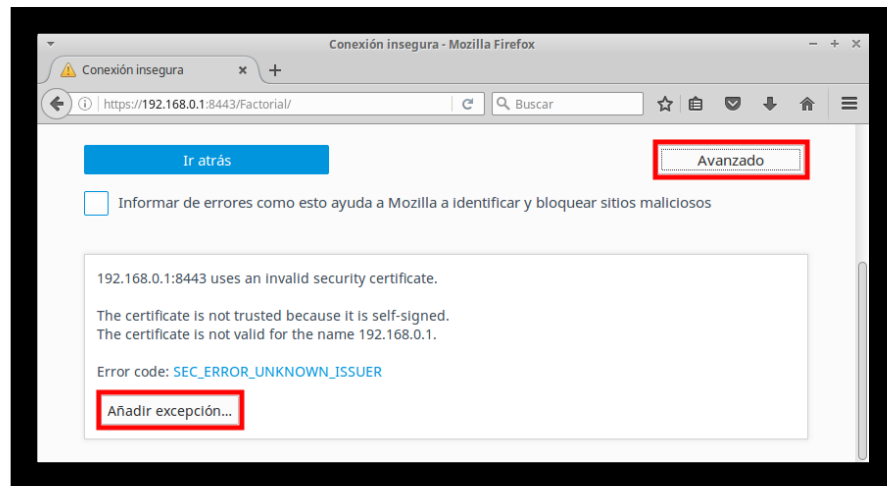
```
sudo service tomcat9 restart
```

Comprobamos que la configuración fue la adecuada, accediendo desde el navegador de la máquina cliente a `https://192.168.0.1:8443/Factorial`. Donde se nos indicará que la configuración no es la correcta, ya que el certificado está firmado por sí mismo.





- Añadimos una excepción y ya podemos acceder.



- Comprobamos que si tratamos de acceder sin SSL también podemos <http://192.168.0.1:8080/Factorial/>
- Configuramos la aplicación para que solo acepte conexiones HTTPS, añadiendo en el descriptor de despliegue modificado en la tarea anterior los siguientes elementos, como hijos de <security-constraint>:



```
<user-data-constraint>
  <transport-guarantee>CONFIDENTIAL</transport-guarantee>
</user-data-constraint>
```

- Y desde el navegador comprobamos ahora que, si tratamos de acceder mediante el protocolo HTTP, obtenemos un error.

## 8. Valves

### SOLUCIÓN

- En la máquina donde hayamos instalado Netbeans, importaremos la carpeta del proyecto Servlet Formulario.
- Configuramos la Valve. Para eso, dentro de este proyecto, creamos el fichero `META-INF/context.xml`, haciendo clic derecho sobre `Web Content/META-INF` y escogiendo `New...→ File` y escribimos las siguientes líneas:

```
<Context>
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="192.168.0.3"/>
</Context>
```

- Eliminamos la aplicación del servidor Tomcat (si ya la habíamos desplegado anteriormente), creamos el fichero WAR y la volvemos a desplegar empleando el método que nos resulte más cómodo (Tomcat Manager, Ant o despliegue manual).
- Comprobamos que la configuración fue la adecuada, accediendo desde el navegador de la máquina cliente a `http://192.168.0.1:8080/Factorial` y comprobamos que podemos acceder. Accedemos desde otra máquina, por ejemplo, desde la máquina Windows 7 y comprobamos que no se permite el acceso (error 403).





**9. Crea una Valve para todo el servidor Tomcat de Ubuntu que sólo permita acceder al ordenador cliente.**

- Modificamos el fichero “/etc/tomcat7/server.xml”
- Dentro de la etiqueta <Engine> añadimos:

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="192.168.0.3"/>
```

- Reiniciamos el Tomcat
- Accedemos a la siguiente dirección y vemos que nos permite acceder.

<http://192.168.0.3:8080>

- Accedemos desde cualquier otro equipo y comprobamos que no podemos acceder.

**10. Crea una Valve para todo el servidor Tomcat de Ubuntu que NO permita acceder al ordenador cliente.**

- Modificamos el fichero “/etc/tomcat7/server.xml”
- Dentro de la etiqueta <Engine> añadimos:

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve" deny="192.168.0.3"/>
```

- Reiniciamos el Tomcat
- Accedemos a la siguiente dirección y vemos que NO nos permite acceder.



<http://192.168.0.3:8080>

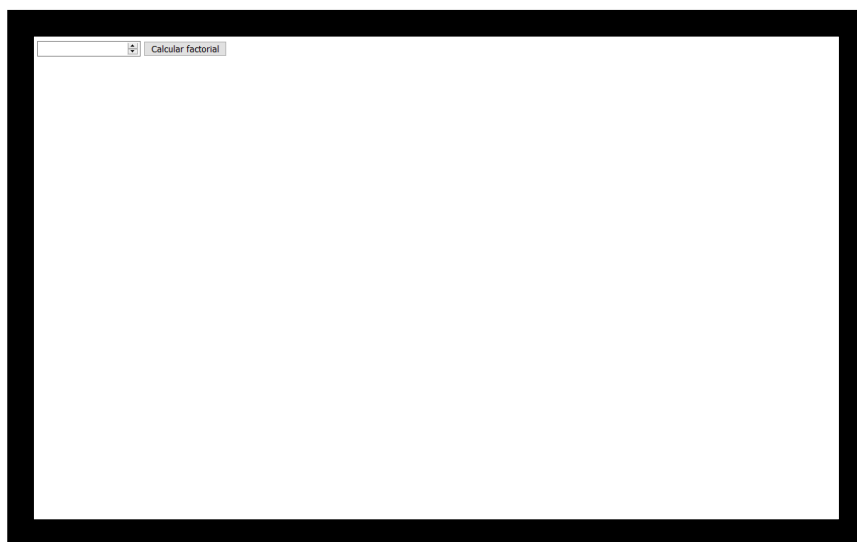
- Accedemos desde cualquier otro equipo y comprobamos que Sí podemos acceder.

#### 11. Para el ejercicio 6 configura:

- a. Un Valve que permita acceder solamente al ordenador cliente
- Editamos o creamos el archivo context.xml en el directorio web/META-INF y añadimos:

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="192.168.0.3"/>
```

- Desplegamos la aplicación <http://192.168.0.3:8080/Ejercicio6>



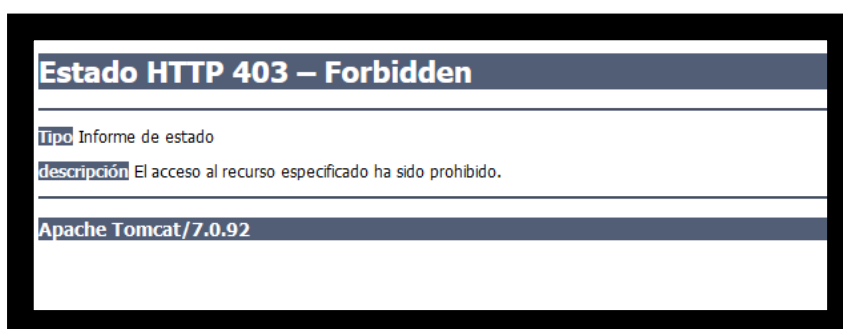
- b. Un Valve que impida el acceso al ordenador cliente



- Editamos o creamos el archivo context.xml en el directorio web/META-INF y añadimos:

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve" deny="192.168.0.3"/>
```

- Desplegamos la aplicación <http://192.168.0.3:8080/Ejercicio6>



## 12. Funcionamiento de Tomcat Web Manager

### SOLUCION

#### a. Crea un usuario con privilegios para acceder a la aplicación

Modificamos el fichero tomcat-users.xml y añadimos un usuario con el rol de manager-gui

```
<role rolename="manager-gui"/>  
<user password="montecastelodaw" roles="manager-gui,admin" username="alumnodaw"/>
```

Reiniciamos el Tomcat

#### b. Despliega una aplicación de las creadas usando Tomcat Web Manager

Abrimos el Tomcat Manager y seleccionamos una de las aplicaciones



### 13. Aplicación ContadorSesiones

#### SOLUCION

- a. Crea un nuevo proyecto llamado ContadorSesiones y añade el Servlet Sesiones.java.
- b. Configura el descriptor de despliegue

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">

  <servlet>
    <servlet-name>Sesiones</servlet-name>
    <servlet-class>Sesiones</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Sesiones</servlet-name>
    <url-pattern>/Sesiones</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
</web-app>
```

- c. Despliega la aplicación
- Accedemos a la URL **<http://localhost:8080/ContadorSesiones/Sesiones>**



## Accesos: 1 en esta sesión

(Identificador de sesión: 90693C832525D3A125EFAE5C06CCBCAD)

(Fecha de creación de la sesión: Wed Jan 29 12:10:22 CET 2020)

(Fecha de último acceso a la sesión: Wed Jan 29 12:10:22 CET 2020)

(Máximo tiempo inactivo de la sesión: 1800 segundos)

[Refrescar](#)

[Refrescar con reescritura de URLs](#)

## Servlet SesionServlet at /ContadorSesiones

- d. Edita el descriptor de despliegue para que la sesión expire en 2 minutos

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">

  <servlet>
    <servlet-name>Sesiones</servlet-name>
    <servlet-class>Sesiones</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Sesiones</servlet-name>
    <url-pattern>/Sesiones</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      2
    </session-timeout>
  </session-config>
</web-app>
```

## 14. Uso de sesiones en las aplicaciones web

### SOLUCION

- Configura el tiempo de vida de las sesiones en 1 minuto.



Incluimos, en el descriptor de fichero los siguientes elementos (como hijos de <web-app>):

```
<session-config>
  <session-timeout>1</session-timeout>
</session-config>
```

Desplegamos de nuevo la aplicación y accedemos a Tomcat Manager para ver que Sí cambió el tiempo de vida de las mismas:

Gestor de Aplicaciones Web de Tomcat

Mensaje: OK

Gestor

Listar Aplicaciones

Ayuda HTML de Gestor

Ayuda de Gestor

Estado de Servidor

Aplicaciones

Trayectoria	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado		true	0	<div>ArrancarPararRecargarReplegar</div> <div>Expirar sesiones sin trabajar &gt; 30 minutos</div>
/Factorial	Ninguno especificado	ServletFormulario	true	0	<div>ArrancarPararRecargarReplegar</div> <div>Expirar sesiones sin trabajar &gt; 30 minutos</div>
/LoginServlet	Ninguno especificado	LoginServlet	true	0	<div>ArrancarPararRecargarReplegar</div> <div>Expirar sesiones sin trabajar &gt; 1 minutos</div>

- Observa que sucede con las sesiones cuando se reinicia el navegador.  
La sesión se cierra, podemos verlo ya que se nos vuelve a pedir usuario y contraseña.
- Con la sesión abierta, apaga Tomcat.

```
sudo service tomcat8 stop
```

- Comprueba que existe el archivo SESSIONS.ser

```
ls /var/lib/tomcat/work/Catalina/localhost/ContadorSesiones
```



- Reinicia Tomcat, ¿qué pasó con la sesión?

```
sudo service tomcat8 stop
```

Si actualizamos el navegador, por ejemplo, vemos que se sigue mostrando el nombre del usuario, por lo que la sesión se recuperó adecuadamente.

- Mata al proceso que ejecuta Tomcat y reinícialo, ¿qué pasó con la sesión?

```
Terminal - administrador@xubuntu16-daw: ~
Archivo Editar Ver Terminal Pestañas Ayuda
administrador@xubuntu16-daw:~$ ps -aux | grep tomcat
tomcat8 2106 44.4 8.1 2139560 166980 ?        Sl    11:50   0:08 /usr/lib/jvm/de
fault-java/bin/java -Djava.util.logging.config.file=/var/lib/tomcat8/conf/logg
g.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -
Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC -Djava.endorsed.dirs=/u
sr/share/tomcat8/endorsed -classpath /usr/share/tomcat8/bin/bootstrap.jar:/usr/s
hare/tomcat8/bin/tomcat-juli.jar -Dcatalina.base=/var/lib/tomcat8 -Dcatalina.hom
e=/usr/share/tomcat8 -Djava.io.tmpdir=/tmp/tomcat8-tomcat8-tmp org.apache.catali
na.startup.Bootstrap start
adminis+ 2146  0.0  0.0  21736  1088 pts/4    S+   11:51   0:00 grep --color=au
to tomcat
administrador@xubuntu16-daw:~$ sudo kill -9 2106
administrador@xubuntu16-daw:~$ sudo service tomcat8 start
administrador@xubuntu16-daw:~$
```

Las sesiones no se guardan, se vuelve a solicitar usuario y contraseña.

## 15. Filtros

### SOLUCION

- Incluimos el archivo FiltroContador.java en nuestro proyecto, en la carpeta Java resources → src → (default package).
- Editamos el descriptor de aplicación `web.xml` para incluir las siguientes líneas (dentro del elemento `<web-app>`):



```
<filter>
  <filter-name>FiltroContador</filter-name>
  <filter-class>FiltroContador</filter-class>
</filter>
<filter-mapping>
  <filter-name>FiltroContador</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

- Desplegamos la aplicación y comprobamos que se van contando las visitas al cambiar entre las páginas de la aplicación.

## 16. Ficheros de registro

### SOLUCION

- Consulta el fichero `/var/lib/tomcat8/conf/server.xml`. ¿Cuál es la Valve que configura la localización y formato de los registros de acceso?

```
<Valve className="org.apache.catalina.valves.AccessLogValve"
directory="logs" prefix="localhost_access_log" suffix=".txt"
pattern="%h %l %u %t &quot;%r&quot; %s %b" />
```

- Lista el directorio de ficheros de registro.

```
ls /var/lib/tomcat8/logs
```

- Abre alguno de los ficheros de registro de acceso y analiza su contenido.

```
192.168.0.1 - - [26/Ene/2020:11:59:29 +0100] " GET / manager HTTP/1.1" 302 -
```





Se muestra la IP desde la que se accedió, la fecha y hora, el tipo de petición, el fichero solicitado, el protocolo y el código de la respuesta.

### 17. Creación de documentación con Javadoc

#### SOLUCION

Una solución posible está en el proyecto Ejercicio17.

Generaremos la documentación en HTML y PDF tal y como se indica en los apartados correspondientes.

### 18. Empleo de herramientas colaborativas de generación de documentación

#### SOLUCION

- Crea una cuenta en Bitbucket.

Cada uno debe crear una cuenta en Bitbucket y activarla siguiendo los pasos explicados en el apartado correspondiente.

- Se debe crear un repositorio en Bitbucket, como se explica en el apartado correspondiente.

Desde Netbeans, debe añadir un nuestro repositorio remoto a un proyecto existente y hacer un push a dicho repositorio.

- Activad la wiki.

Si no se activó en opciones avanzadas al crear el repositorio, debe hacerse desde el apartado Settings→Wiki

- Editad la wiki de forma colaborativa.

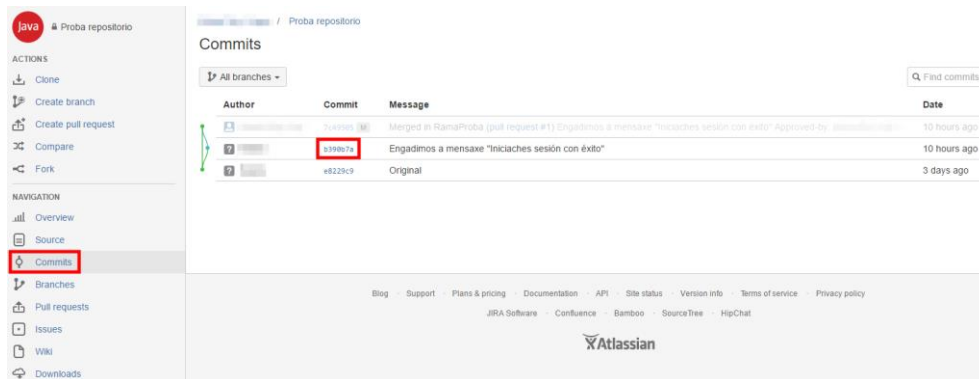
Se hará tal y como se explica en el apartado correspondiente, por ejemplo, podemos escribir el siguiente código:

```
# Prueba de repositorio #  
  
Vamos a probar:  
  
* La wiki  
* Las discusiones in-line  
* Los pull requests
```



- Probad las discusiones in-line.

Accedemos a alguno de los commits del proyecto, haciendo clic en Commits y en alguno de los identificadores de los commits:



A continuación, hacemos clic sobre este icono en la línea que queramos escribir el comentario. Si queremos responder a él, solo tenemos que hacer clic en Reply.



- Probad el pull request.

En Netbeans, se creará una nueva rama, se hará una modificación y, a continuación, un commit y un push al repositorio.