

## CICLO : DESARROLLO APLICACIONES WEB-DISTANCIA

### MÓDULO : DESPLIEGUE DE APLICACIONES WEB

ALUMNO : DAVID MEDINA GARCIA

#### TAREA : DAW03

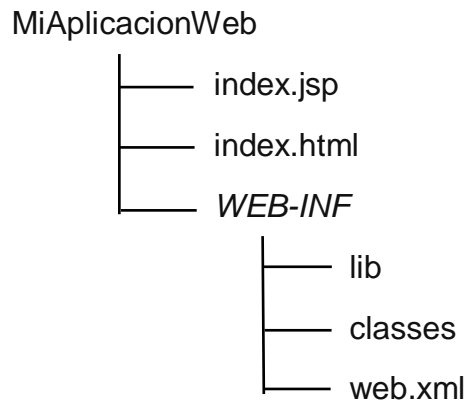
1. Una aplicación web puede ser desplegada en diferentes servidores web manteniendo su funcionalidad y sin ningún tipo de modificación en su código debido a la especificación servlet 2.2, ¿cuál es la estructura de directorios que debe tener?

Ha de tener un **directorio principal o raíz** que contendrá los **ficheros estáticos** y **JSP's**.

En su interior ha de existir una carpeta denominada **WEB-INF** que contendrá el fichero **web.xml** encargado de configurar la aplicación.

Dentro de esta carpeta tendremos dos **supcarpetas**, una denominada **classes** que contiene los ficheros compilados(Servlets, beans) y otra denominada **lib** que tendrá las librerías adicionales.

Al mismo nivel de **WEB-INF** estará el resto de carpetas para ficheros estáticos.



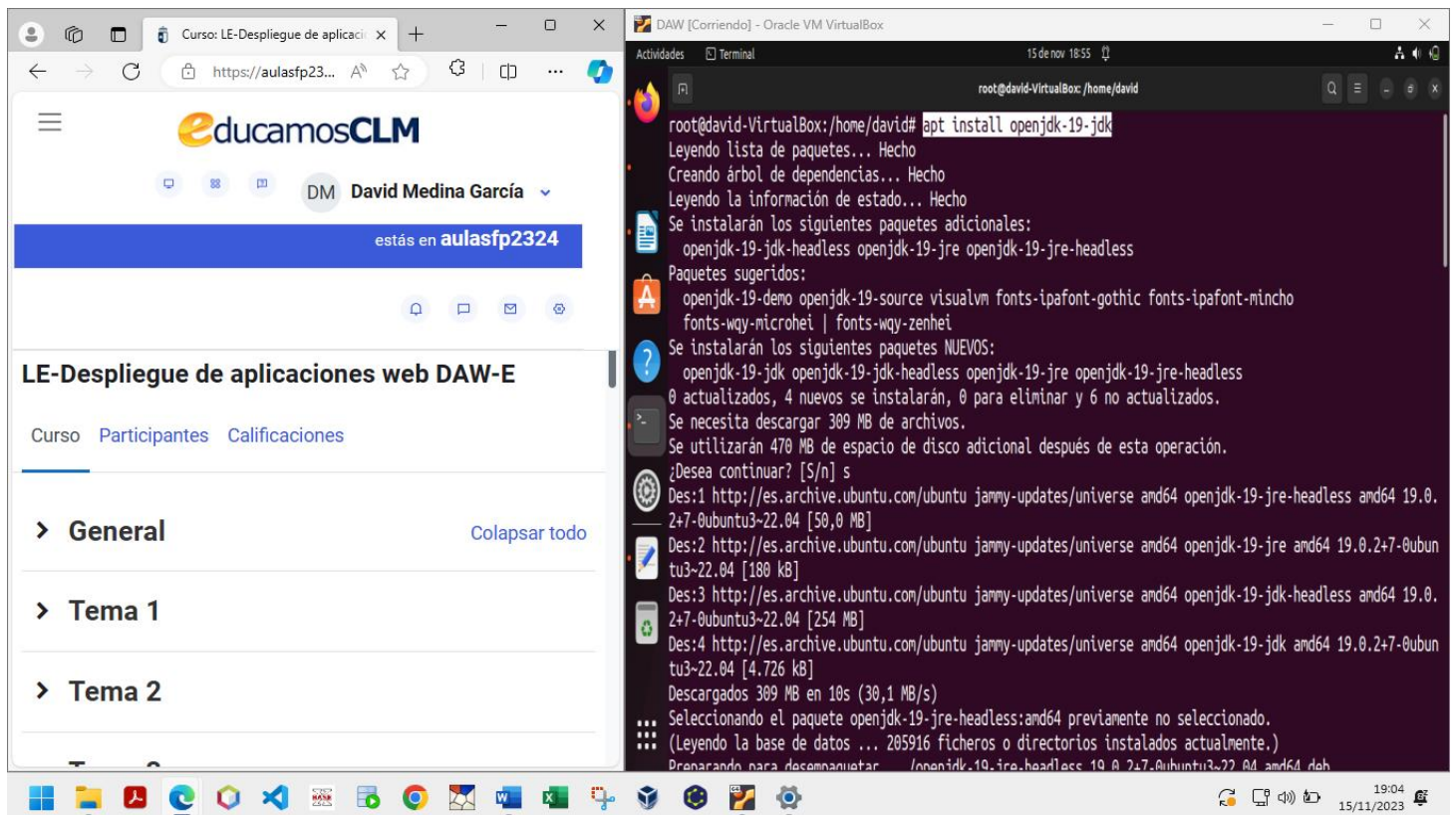
2. Ant se basa en ficheros XML, normalmente configuramos el trabajo a hacer con nuestra aplicación en un fichero llamado build.xml. Indica alguna de las etiquetas con las que podemos formar el contenido de este archivo, así como la función de cada una de ellas.
  - **project**: este es el elemento raíz del fichero XML, y como tal, solamente puede haber uno en todo el fichero, el que se corresponde a nuestra aplicación Java.
  - **target**: Es un conjunto de tareas que queremos aplicar a nuestra aplicación en algún momento. Se puede hacer que unos objetivos dependan de otros, de forma que eso lo trate Ant automáticamente.
  - **task**: Es un código ejecutable que aplicaremos a nuestra aplicación, y que puede contener distintas propiedades (como por ejemplo el **classpath**). Ant incluye ya muchas básicas, como compilación y eliminación de ficheros temporales, pero podemos extender este mecanismo si nos hace falta. Luego veremos algunas de las disponibles.
  - **property**: Es simplemente algún parámetro (en forma de par nombre-valor) que necesitamos para procesar nuestra aplicación, como el nombre del compilador, etc. Ant incluye ya las más básicas, como son BaseDir para el directorio base de nuestro proyecto, ant.file para el path absoluto del fichero build.xml, y ant.java.version para la versión de la JVM.

3. Dispones de una máquina que cuenta con el sistema operativo Ubuntu 18.04 LTS o posterior, recientemente actualizado, en la que está el entorno de red configurado y, además, dispones de conexión a Internet y estás trabajando con la cuenta del usuario root . Indica cada uno de los pasos, y comandos implicados en ellos, para conseguir hacer lo siguiente:

- Instalar el JDK 8.

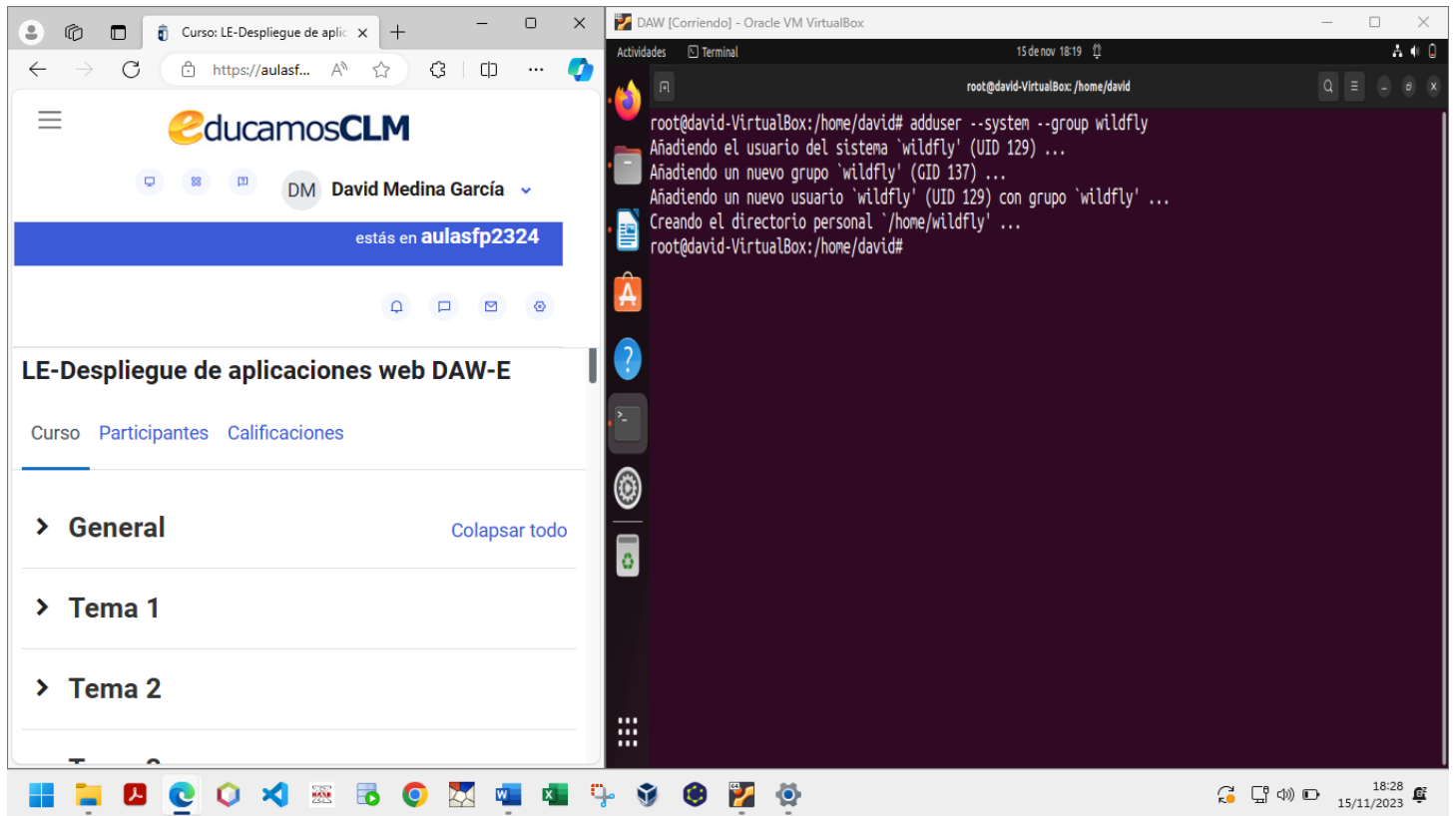
Instalamos una versión más reciente de java.

**apt install openjdk-19-jdk**



- Crear usuario para WildFly.

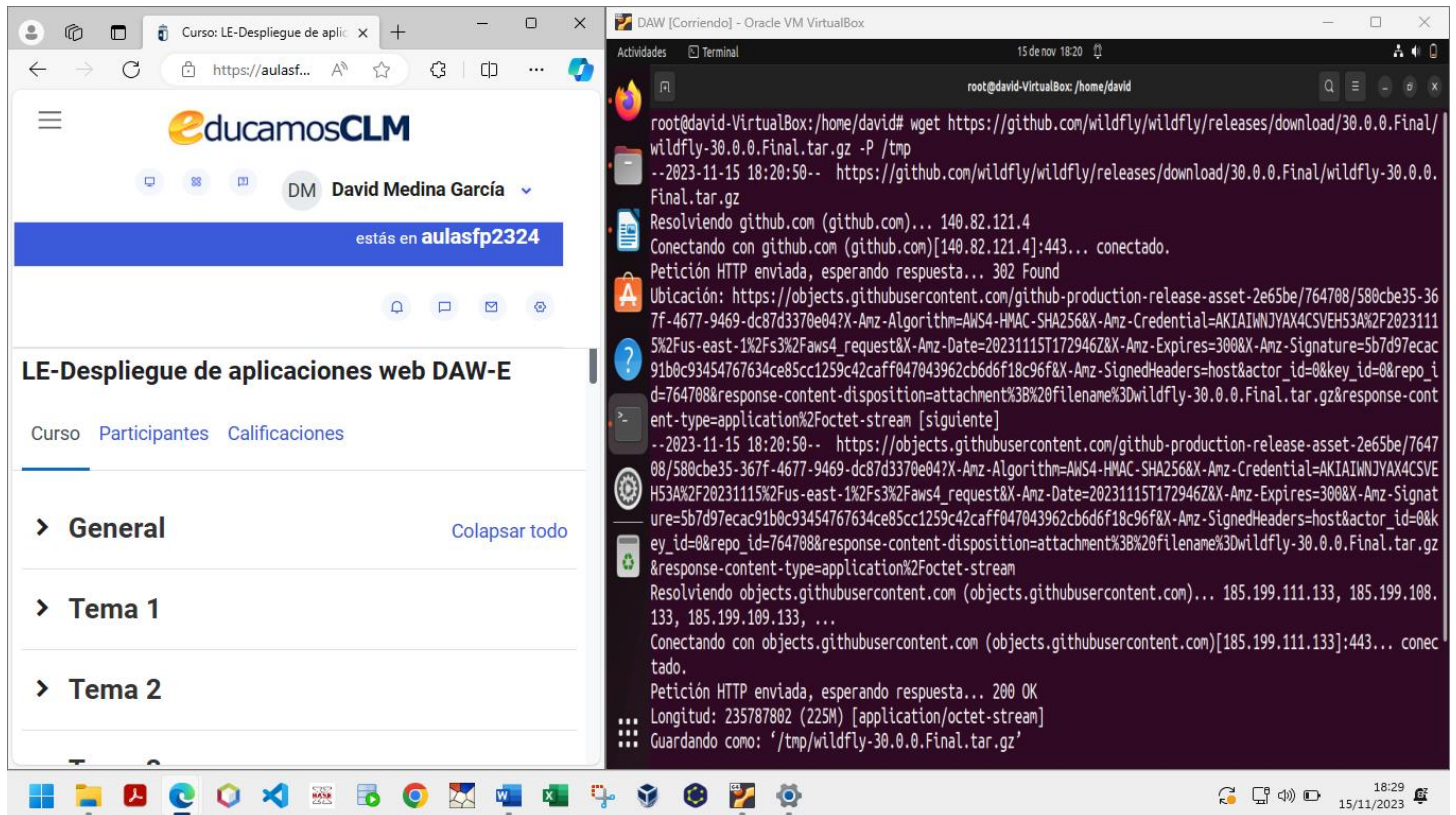
Creamos un usuario y un grupo llamados **wildfly** en el sistema con el comando **adduser --system --group wildfly**.



- **Descargar e instalar WildFly 19.0.0 Final.**

Descargamos la versión más reciente de **wildfly(30.0.0)** en el directorio **/tmp**

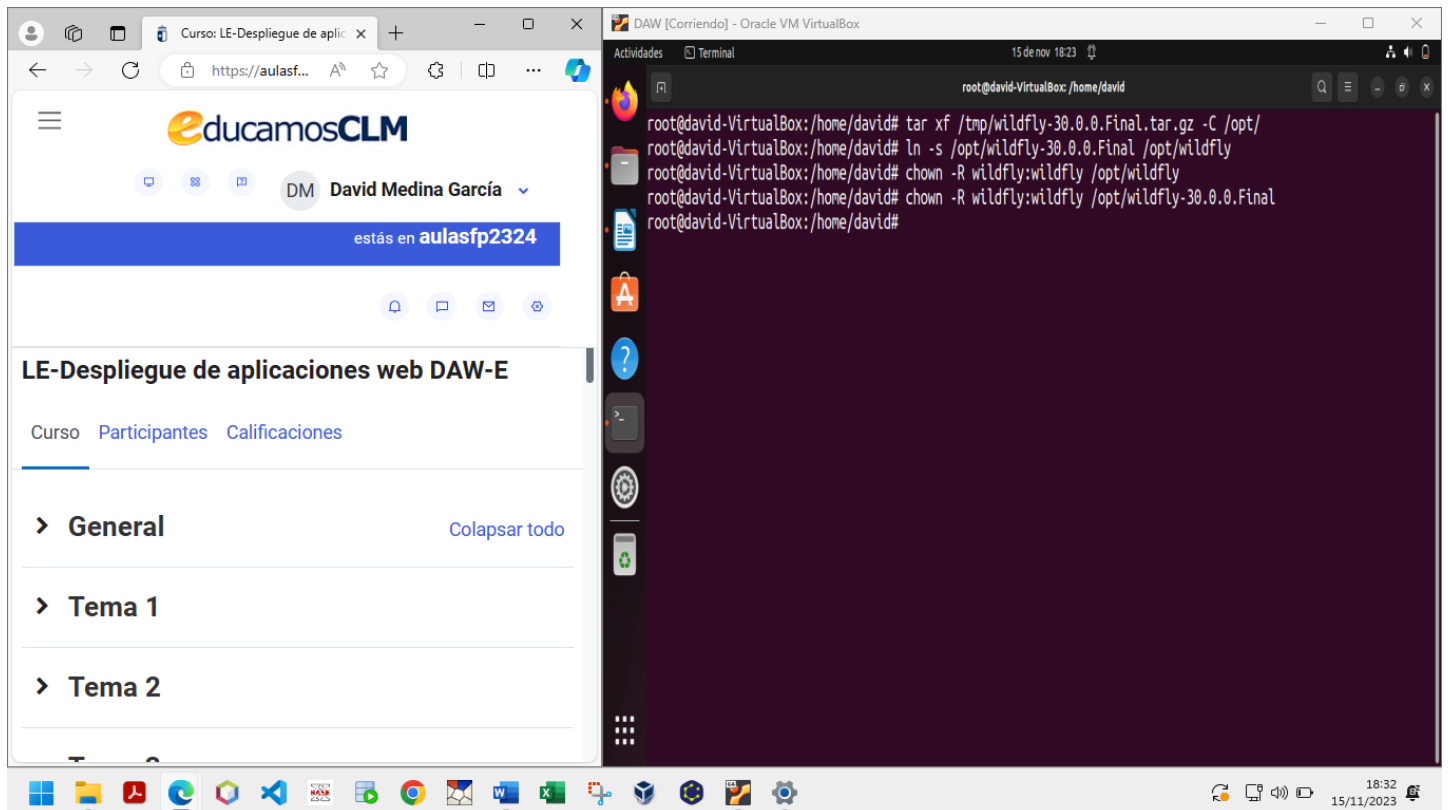
**wget** <https://github.com/wildfly/wildfly/releases/download/30.0.0.Final/wildfly-30.0.0.Final.tar.gz> -P /tmp



Se descomprime el archivo en la carpeta **/opt**  
**tar xf /tmp/wildfly-30.0.0.Final.tar.gz -C /opt**

Creamos el enlace simbólico para **WildFly**  
**ln -s /opt/wildfly-30.0.0.Final /opt/wildfly**

Le pasamos la propiedad de las carpetas al usuario **wildfly**  
**chown -R wildfly:wildfly /opt/wildfly**  
**chown -R wildfly:wildfly /opt/wildfly-30.0.0.Final**



- **Configurar systemd y el archivo wildfly.conf.**

Creamos un archivo de servicio en la carpeta system  
**nano /etc/systemd/system/wildfly.service**

Le agregamos el siguiente contenido

**[Unit]**

**Description=The WildFly Application Server**

**After=syslog.target network.target**

**[Service]**

**User=wildfly**

**Group=wildfly**

**ExecStart=/opt/wildfly/bin/standalone.sh -c standalone-full.xml**

**ExecStop=/opt/wildfly/bin/jboss-cli.sh --connect command=:shutdown**

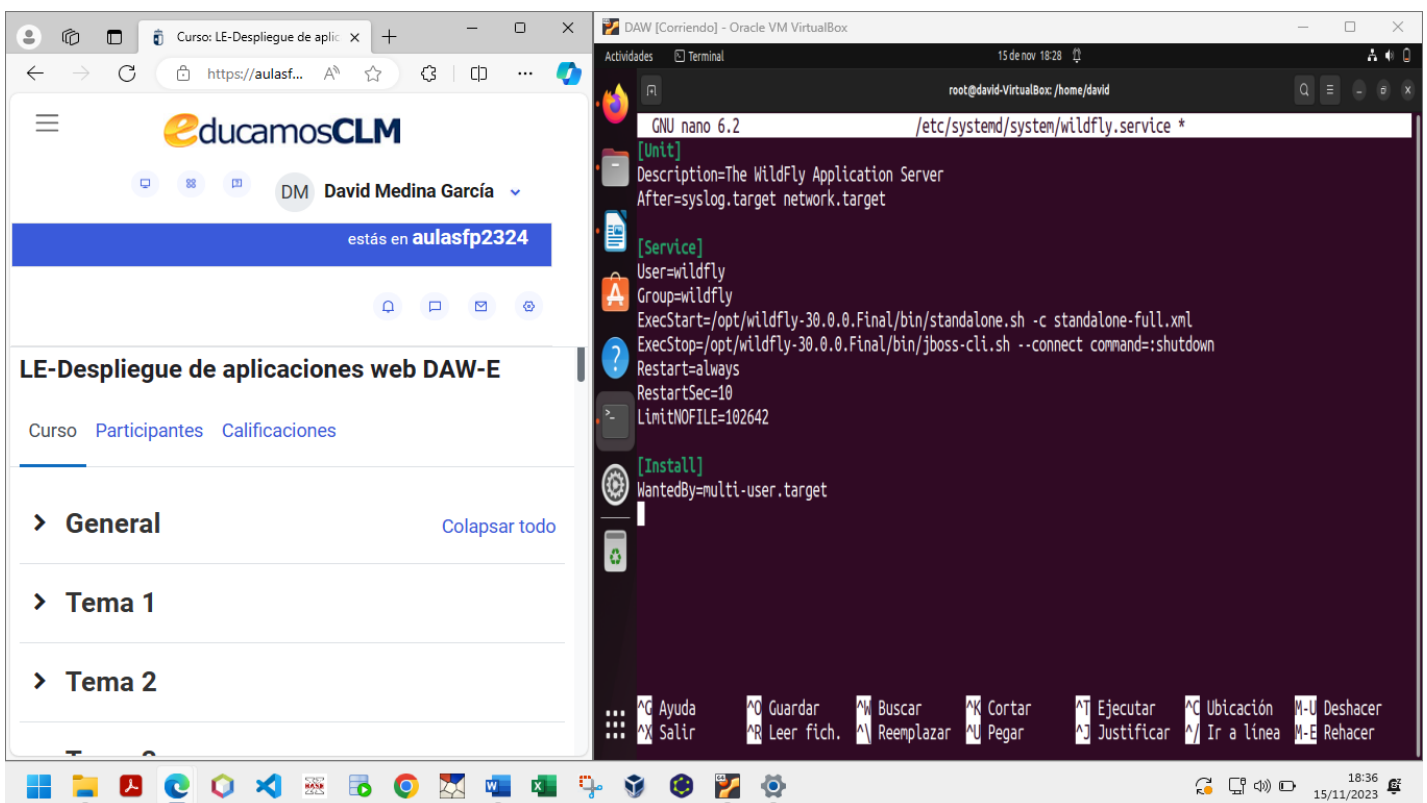
**Restart=always**

**RestartSec=10**

**LimitNOFILE=102642**

**[Install]**

**WantedBy=multi-user.target**





Recargamos **systemd**  
**systemctl daemon-reload**

Habilitamos el servicio.  
**systemctl enable wildfly**

Iniciamos el servicio.  
**systemctl start wildfly**

Comprobamos su estado.  
**systemctl status wildfly**

The screenshot shows a web browser on the left and a terminal window on the right. The browser displays the 'aulasfp2324' page for 'David Medina García' under the 'LE-Despliegue de aplicaciones web DAW-E' course. The terminal window shows the following commands and output:

```
root@david-VirtualBox:/opt# systemctl enable wildfly
Created symlink /etc/systemd/system/multi-user.target.wants/wildfly.service → /etc/systemd/system/wildfly.service.
root@david-VirtualBox:/opt# systemctl start wildfly
root@david-VirtualBox:/opt# systemctl status wildfly
● wildfly.service - The WildFly Application Server
   Loaded: loaded (/etc/systemd/system/wildfly.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-11-15 19:05:37 CET; 7s ago
     Main PID: 20635 (standalone.sh)
       Tasks: 130 (limit: 6960)
      Memory: 299.0M
         CPU: 8.847s
    CGroup: /system.slice/wildfly.service
            └─20635 /bin/sh /opt/wildfly/bin/standalone.sh -c standalone-full.xml
              └─20756 java "-D[Standalone]" -Xms64m -Xmx512m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256M

nov 15 19:05:43 david-VirtualBox standalone.sh[20756]: 19:05:43,807 INFO [org.jboss.as.controller] (Cont>
nov 15 19:05:43 david-VirtualBox standalone.sh[20756]: WFLYCTL0184: New missing/unsatisfied dependenci>
nov 15 19:05:43 david-VirtualBox standalone.sh[20756]: service jboss.http-upgrade-registry.default>
nov 15 19:05:43 david-VirtualBox standalone.sh[20756]: service org.wildfly.undertow.listener.http-u>
nov 15 19:05:43 david-VirtualBox standalone.sh[20756]: WFLYCTL0186: Services which failed to start: >
nov 15 19:05:43 david-VirtualBox standalone.sh[20756]: WFLYCTL0448: 2 additional services are down due to>
nov 15 19:05:43 david-VirtualBox standalone.sh[20756]: 19:05:43,898 INFO [org.jboss.as.server] (Controll>
nov 15 19:05:43 david-VirtualBox standalone.sh[20756]: 19:05:43,901 INFO [org.jboss.as] (Controller Boot>
nov 15 19:05:43 david-VirtualBox standalone.sh[20756]: 19:05:43,902 INFO [org.jboss.as] (Controller Boot>
nov 15 19:05:43 david-VirtualBox standalone.sh[20756]: 19:05:43,904 ERROR [org.jboss.as] (Controller Boot>
lines 1-21/21 (END)
```

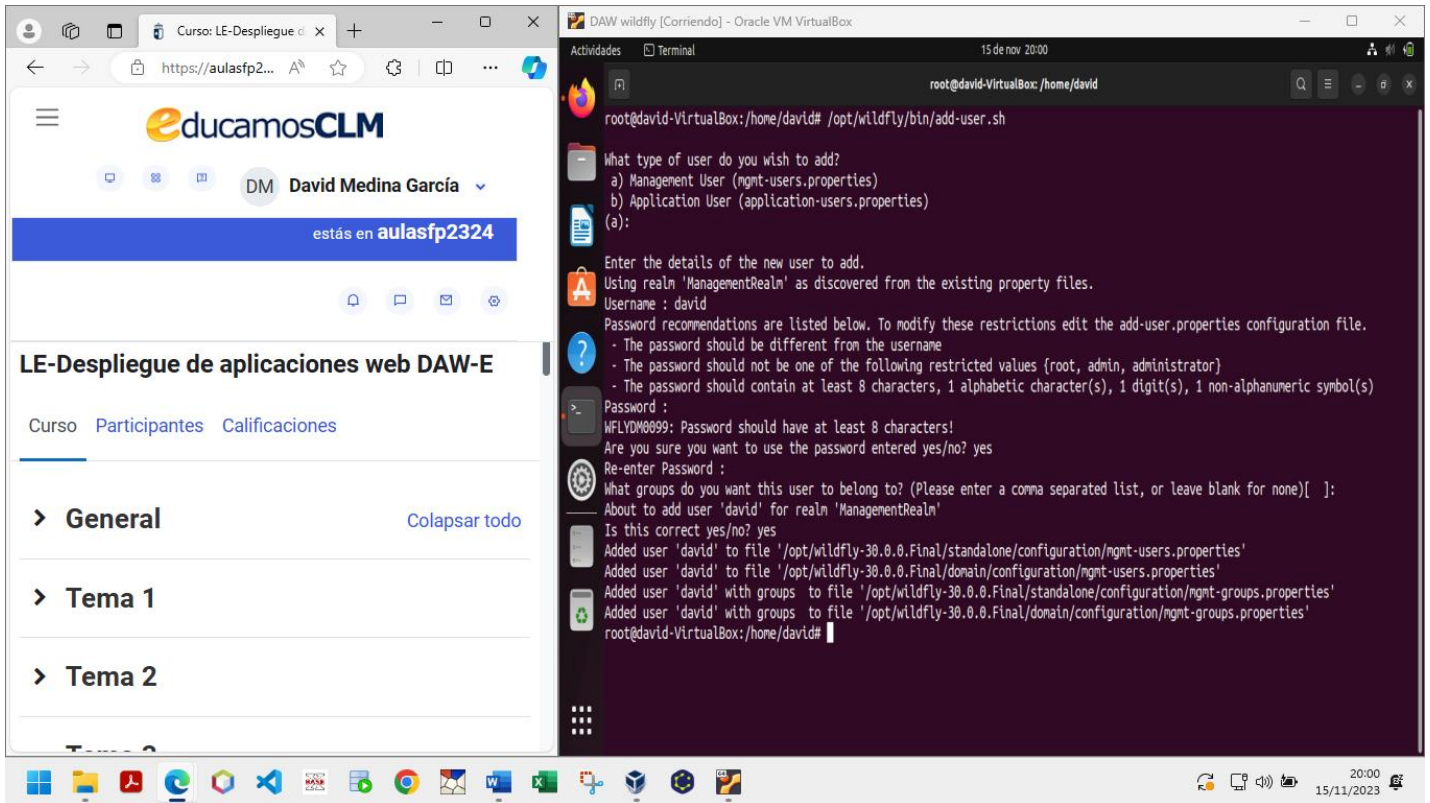
## ○ Configurar la autenticación de Wildfly.

Añadimos un usuario usando el script **add-user.sh**:

**/opt/wildfly/bin/add-user.sh**

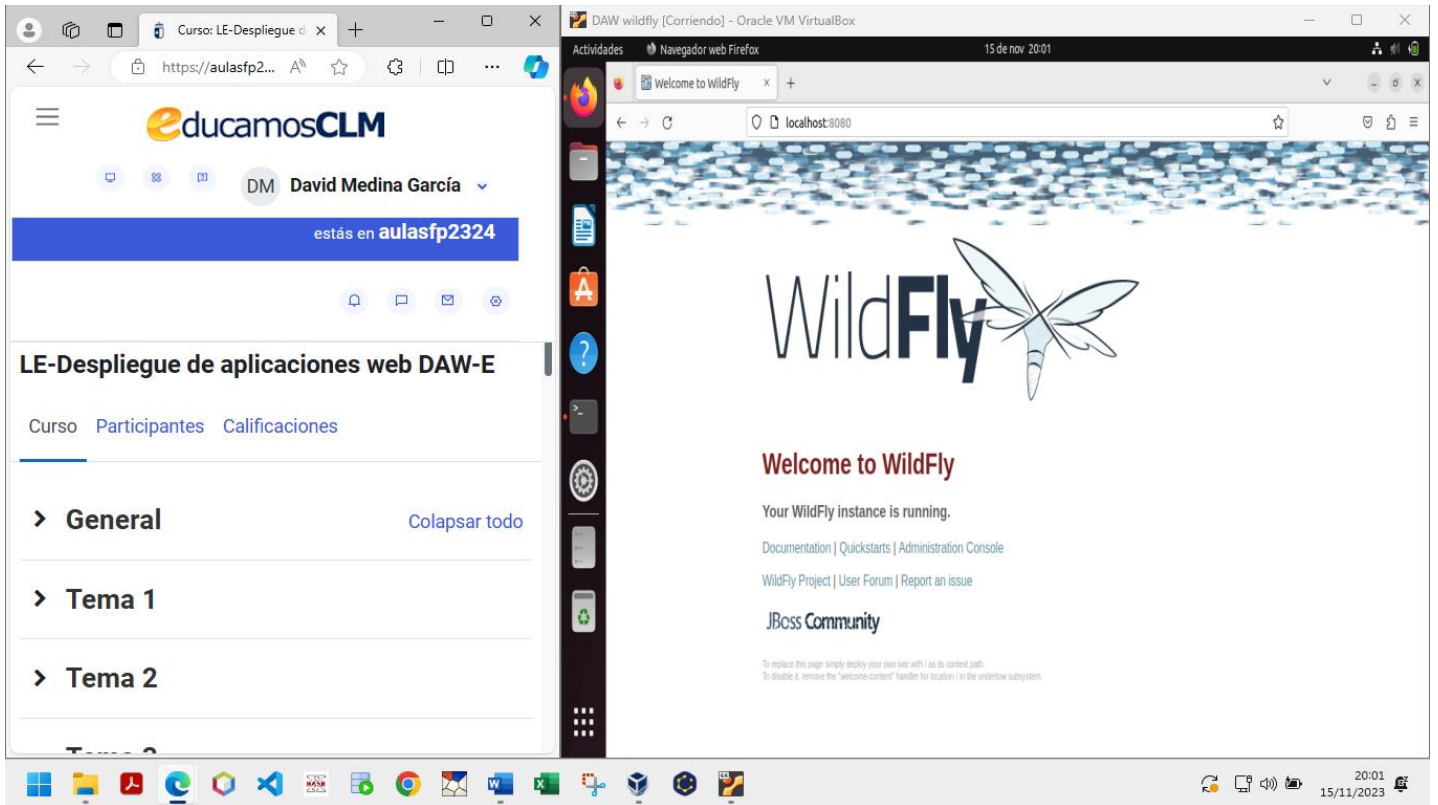
Seleccionamos la opción a (**Management user**)

Añadimos el **usuario** y la **contraseña**





Comprobamos en el navegador accediendo a esta dirección **localhost:8080**



Accedemos en el navegador con esta dirección **localhost:9990** a la consola de administración de **Wildfly**.  
Nos pedirá el usuario y contraseña que creamos antes.

