

# TP7

## 1- Poniendo en funcionamiento Jenkins

- Crear una imagen de Docker que se base en la imagen oficial de Jenkins y que tenga instalado .NET Core. Crear un archivo llamado Dockerfile.jenkins con el siguiente contenido:

```
FROM jenkins/jenkins:lts

USER root

# Instala dependencias necesarias
RUN apt-get update && apt-get install -y \
    apt-transport-https \
    software-properties-common \
    wget

# Agrega el repositorio de Microsoft y actualiza
RUN wget -q https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb && \
    dpkg -i packages-microsoft-prod.deb && \
    apt-get update

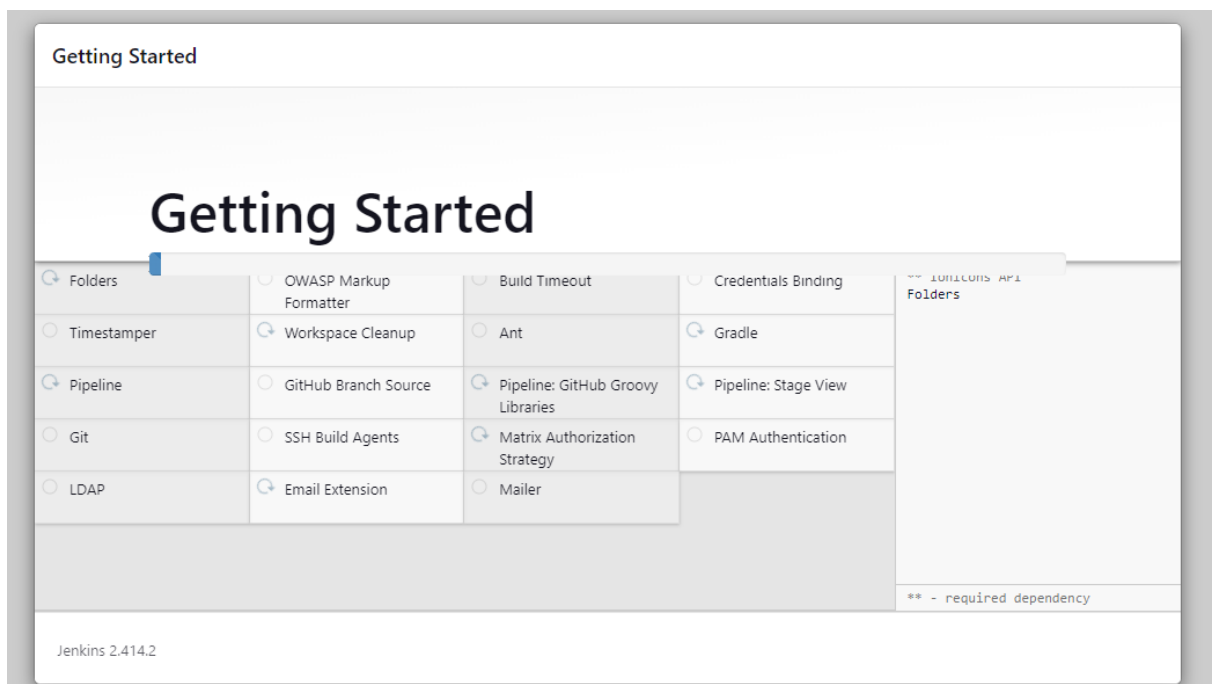
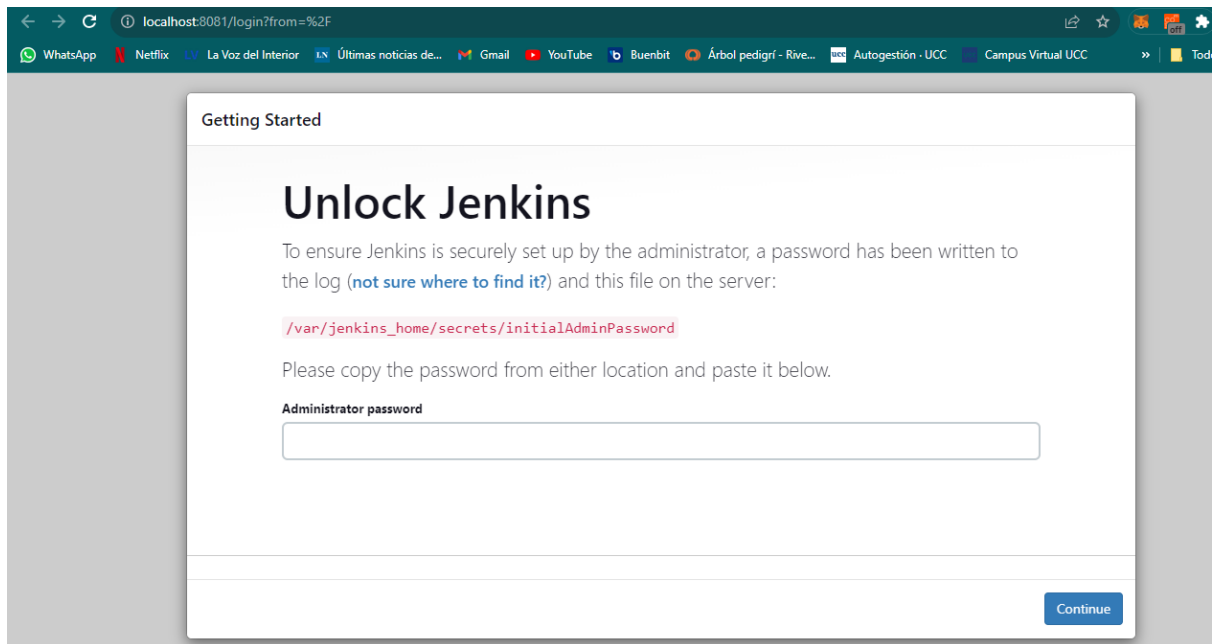
# Instala el SDK de .NET Core
RUN apt-get install -y dotnet-sdk-7.0

USER jenkins
```

- Desde la misma ubicación donde tengas el archivo Dockerfile personalizado, ejecuta el siguiente comando para construir una nueva imagen de Docker con .NET Core y Jenkins. Esto creará una nueva imagen de Docker llamada jenkins-with-dotnetcore:

```
PS D:\Santiago\Ingenieria en Sistemas\4to Año\2do Semestre\Ingenieria de Software III\Practico\ing-software-3\TP7> docker build -t jenkins2 -f dockerfile .
[+] Building 121.6s (8/8) FINISHED
=> [internal] load build definition from dockerfile                                0.0s
=> => transferring dockerfile: 562B                                              0.0s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/jenkins/jenkins:lts                  0.0s
=> [1/4] FROM docker.io/jenkins/jenkins:lts                                     0.3s
=> [2/4] RUN apt-get update && apt-get install -y apt-transport-https software-properties-common wget 67.5s
=> [3/4] RUN wget -q https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb && dpkg -i pack 7.5s
=> [4/4] RUN apt-get install -y dotnet-sdk-7.0                                  40.7s
=> exporting to image                                                            5.6s
=> => exporting layers                                                            5.6s
=> => writing image sha256:ad6d52f7192b25708b344db3f8a1373ea4ceb7b5de94a245e46510596f1d51a8 0.0s
=> => naming to docker.io/library/jenkins2                                       0.0s
```

```
PS D:\Santiago\Ingenieria en Sistemas\4to Año\2do Semestre\Ingenieria de Software III\Practico\ing-software-3\TP7> docker run -d -p 8080:8080 -p 50000:50000 -
-name jenkins -v c:\jenkins\var\jenkins_home jenkins2
5d3d27771b4cbfa27814164f6f8b9f038d8acb88554c7f6988d8afe5a61fc23b2
```



Getting Started

# Create First Admin User

Usuario

Contraseña

Confirma la contraseña







Nombre completo

Jenkins 2.414.2

[Skip and continue as admin](#) [Save and Continue](#)

### 3- Creando el primer Job de estilo libre


Crear un nuevo item, del tipo estilo libre con nombre **first-job**

 **Jenkins**      **Santiago Riveros Salomon**  [Desconectar](#)


Panel de Control > Todo >

**Enter an item name**

» Required field

 **Crear un proyecto de estilo libre**



Esta es la característica principal de Jenkins. la de ejecutar el proyecto combinando cualquier tipo de repositorio de software (SCM) con cualquier modo de construcción o ejecución (make, ant, mvn, rake, script ...). Por tanto se podrá tanto compilar y empaquetar software, como ejecutar cualquier proceso que requiera monitorización.


 **Pipeline**

Gestiona actividades de larga duración que pueden abarcar varios agentes de construcción. Apropiado para construir pipelines (conocidas anteriormente como workflows) y/o para la organización de actividades complejas que no se pueden articular facilmente con tareas de tipo freestyle.

Una vez creado el job, en la sección Build Steps seleccionamos **Ejecutar linea de comandos (shell)** y escribimos:

## Build Steps

 Ejecutar línea de comandos (shell) 




Comando

Visualizar [la lista de variables de entorno disponibles](#)

```
# Obtén la fecha y hora actual en formato personalizado
current_datetime=$(date +%Y-%m-%d %H:%M:%S)

# Imprime la fecha y hora actual utilizando el comando echo
echo "La fecha y hora actual es: $current_datetime"
```

Avanzado 



## Salida de consola

```
Started by user Santiago Riveros Salomon
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/first-job
[first-job] $ /bin/sh -xe /tmp/jenkins13817233889982423089.sh
+ date +%Y-%m-%d %H:%M:%S
+ current_datetime=2023-10-06 00:31:47
+ echo La fecha y hora actual es: 2023-10-06 00:31:47
La fecha y hora actual es: 2023-10-06 00:31:47
Finished: SUCCESS
```

La respuesta que proporcionaste parece ser la salida del registro de una ejecución de trabajo en Jenkins. Aquí tienes una breve descripción de lo que está sucediendo en esta respuesta:

1. "Started by user Santiago Riveros Salomon": Indica que el trabajo (job) en Jenkins fue iniciado por el usuario llamado Santiago Riveros Salomon.
2. "Running as SYSTEM": Esto muestra que el trabajo se está ejecutando como el usuario del sistema (SYSTEM) de Jenkins.
3. "Building in workspace /var/jenkins\_home/workspace/first-job": Indica la ubicación del directorio de trabajo en el que se está ejecutando el trabajo. En este caso, el trabajo se está ejecutando en el directorio "/var/jenkins\_home/workspace/first-job".
4. "[first-job] \$ /bin/sh -xe /tmp/jenkins13817233889982423089.sh": Esta línea muestra el comando que Jenkins está ejecutando como parte de este trabajo. En este caso, se está ejecutando un script shell ubicado en "/tmp/jenkins13817233889982423089.sh".
5. Luego, se ejecuta el comando "date +%Y-%m-%d %H:%M:%S" para obtener la fecha y hora actual y se almacena en la variable "current\_datetime".
6. Después, se imprime en pantalla la frase "La fecha y hora actual es: 2023-10-06 00:31:47", que incluye la fecha y hora obtenida en el paso anterior.
7. "Finished: SUCCESS": Finalmente, se indica que la ejecución del trabajo ha finalizado con éxito (SUCCESS). Esto significa que el trabajo se ejecutó correctamente sin errores.

#### 4- Creando el primer Pipeline Job

- Crear un nuevo ítem, del tipo Pipeline con nombre **hello-world**

**Enter an item name**

hello-world

» Required field

**Crear un proyecto de estilo libre**

Esta es la característica principal de Jenkins, la de ejecutar el proyecto combinando cualquier tipo de repositorio de software (SCM) con cualquier modo de construcción o ejecución (make, ant, mvn, rake, script ...). Por tanto se podrá tanto compilar y empaquetar software, como ejecutar cualquier proceso que requiera monitorización.

**Pipeline**

Gestiona actividades de larga duración que pueden abarcar varios agentes de construcción. Apropiado para construir pipelines (conocidas anteriormente como workflows) y/o para la organización de actividades complejas que no se pueden articular fácilmente con tareas de tipo freestyle.

**Crear un proyecto multi-configuración**

Adecuado para proyectos que requieran un gran número de configuraciones diferentes, como testear en múltiples entornos, ejecutar sobre plataformas concretas, etc.

- Una vez creado el job, en la sección Pipeline seleccionamos **try sample Pipeline** y luego **Hello World**

Definition

Pipeline script

Script ?

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Hello') {
6       steps {
7         echo 'Hello World'
8       }
9     }
10  }
11 }
12
```

Hello World

☒ Use Groovy Sandbox ?



## Salida de consola

```
Started by user Santiago Riveros Salomon
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/hello-world
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

La respuesta que proporcionaste parece ser la salida del registro de una ejecución de un trabajo de Jenkins que utiliza un pipeline declarativo. Aquí tienes una breve descripción de lo que está sucediendo en esta respuesta:

1. "Started by user Santiago Riveros Salomon": Indica que el trabajo (pipeline) en Jenkins fue iniciado por el usuario llamado Santiago Riveros Salomon.
2. "[Pipeline] Start of Pipeline": Esto marca el inicio del pipeline.
3. "[Pipeline] node": Indica que el pipeline se está ejecutando en un nodo de Jenkins. En este caso, se ejecuta en un nodo llamado "Jenkins" en el directorio "/var/jenkins\_home/workspace/hello-world".
4. "[Pipeline] {"": Indica el comienzo del bloque principal del pipeline.
5. "[Pipeline] stage": Define una etapa dentro del pipeline. En este caso, la etapa se llama "Hello".
6. "[Pipeline] { (Hello)": Marca el inicio de la etapa llamada "Hello".
7. "[Pipeline] echo": Ejecuta un comando para imprimir en la consola. En este caso, se imprime "Hello World", lo que indica que esta es la tarea que se está realizando en esta etapa.
8. "[Pipeline] }": Marca el final de la etapa "Hello".
9. "[Pipeline] // stage": Indica el final de la etapa en el pipeline.
10. "[Pipeline] }": Marca el final del bloque principal del pipeline.
11. "[Pipeline] // node": Indica el final de la ejecución en el nodo.
12. "[Pipeline] End of Pipeline": Marca el final del pipeline.
13. "Finished: SUCCESS": Finalmente, se indica que la ejecución del pipeline ha finalizado con éxito (SUCCESS). Esto significa que el pipeline se ejecutó correctamente sin errores.

### **5- Creando un Pipeline Job con Git**

### Enter an item name

» Required field



#### Crear un proyecto de estilo libre

Esta es la característica principal de Jenkins, la de ejecutar el proyecto combinando cualquier tipo de repositorio de software (SCM) con cualquier modo de construcción o ejecución (make, ant, mvn, rake, script ...). Por tanto se podrá tanto compilar y empaquetar software, como ejecutar cualquier proceso que requiera monitorización.



#### Pipeline

Gestiona actividades de larga duración que pueden abarcar varios agentes de construcción. Apropiado para construir pipelines (conocidas anteriormente como workflows) y/o para la organización de actividades complejas que no se pueden articular fácilmente con tareas de tipo freestyle.



#### Crear un proyecto multi-configuración

Adecuado para proyectos que requieran un gran número de configuraciones diferentes, como testear en múltiples entornos, ejecutar

### Script ?

```
1 pipeline {
2   agent any
3
4
5   stages {
6     stage('Build') {
7       steps {
8         // Get some code from a GitHub repository
9         git branch: 'main', url: 'https://github.com/ingsoft3ucc/SimpleWebAPI'
10      }
11    }
12  }
13 }
14 }
```

try sample Pipeline... ▼

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

## 6- Utilizando nuestros proyectos.

### Enter an item name

» Required field



#### Crear un proyecto de estilo libre

Esta es la característica principal de Jenkins, la de ejecutar el proyecto combinando cualquier tipo de repositorio de software (SCM) con cualquier modo de construcción o ejecución (make, ant, mvn, rake, script ...). Por tanto se podrá tanto compilar y empaquetar software, como ejecutar cualquier proceso que requiera monitorización.

Ponemos las credenciales que se piden:

### Jenkins Credentials Provider: Jenkins

#### Add Credentials

Domain  
Global credentials (unrestricted) ▼

Kind  
Username with password ▼

Scope ?  
Global (Jenkins, nodes, items, all child items, etc) ▼

Username ?  
SantiRivSal

Cargamos el Repositorio

• Git ?

Repositories ?

Repository URL ?

https://github.com/SantiagoRivSal/ing-software-3.git

Credentials ?

SantiRivSal/\*\*\*\*\* ▼

Add ▼

Avanzado ▼

## **6.2- Monitorear Cambios en el repo de GitHub**

Configurar un "Pull" cada 1 minuto



- ☐ Lanzar ejecuciones remotas (ejem: desde 'scripts') ?
- ☐ Construir tras otros proyectos ?
- ☐ Ejecutar periódicamente ?
- ☐ GitHub hook trigger for GITScm polling ?
- ☒ Consultar repositorio (SCM) ?

Programador ?

H/1 \* \* \* \*

Would last have run at Sunday, October 8, 2023 at 8:05:41 PM Coordinated Universal Time; would next run at Sunday, October 8, 2023 at 9:05:41 PM Coordinated Universal Time.  
[\(mostrar detalles\)](#)

## Ejecutar periódicamente

- ☒ Ejecutar periódicamente ?

Programador ?

H/1 \* \* \* \*

Would last have run at Sunday, October 8, 2023 at 8:05:47 PM Coordinated Universal Time; would next run at Sunday, October 8, 2023 at 9:05:47 PM Coordinated Universal Time.

Un webhook es una forma de que una aplicación web envíe automáticamente datos o información a otra aplicación web en tiempo real cuando ocurren eventos específicos. La implementación de un webhook generalmente requiere el acceso y la capacidad de configurar un servidor web que pueda recibir solicitudes entrantes desde una fuente externa, como una plataforma de terceros o un servicio en línea.

La diferencia principal entre el uso de un webhook y ejecutar una acción periódicamente radica en el enfoque de la sincronización de eventos o acciones en una aplicación o sistema.