

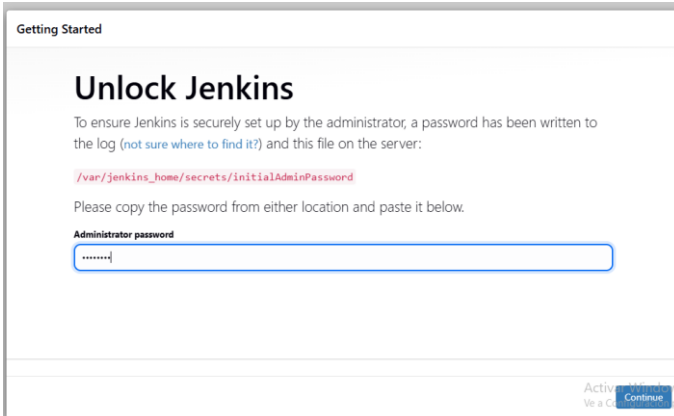
Practica Jenkins

1. Descargar la imagen Docker de Jenkins

`docker pull jenkins/jenkins:lts-jdk17`

2. Crear el contenedor con los mismos puertos por defecto

Ingresar localhos:800



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/jenkins_home/secrets/initialAdminPassword`

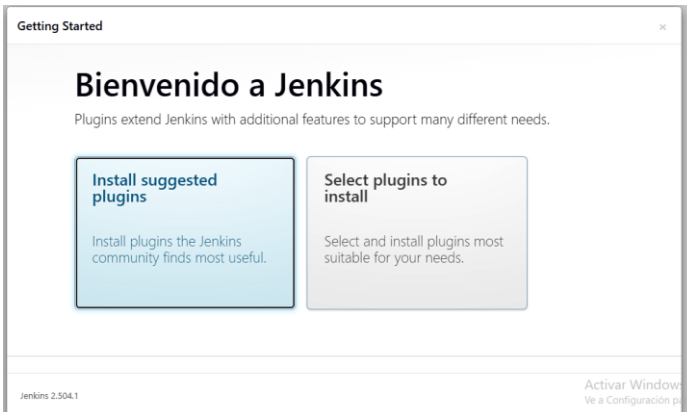
Please copy the password from either location and paste it below.

Administrator password

Activar Windows
Ver a Configuración para...

Continue

3. Copiar el password de la ruta especificada debería ser
a6511c5f189542c1ab6f36ff86a89ae9
4. Instalar los plugins sugeridos



Getting Started

Bienvenido a Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

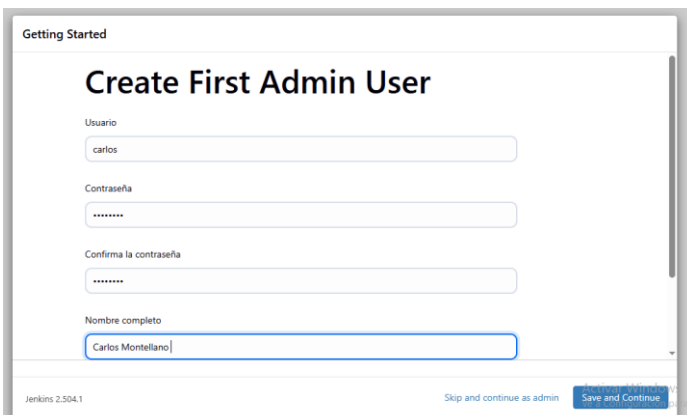
Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.504.1

Activar Windows
Ver a Configuración para...

5. Crear usuario



Getting Started

Create First Admin User

Usuario

Contraseña

Confirma la contraseña

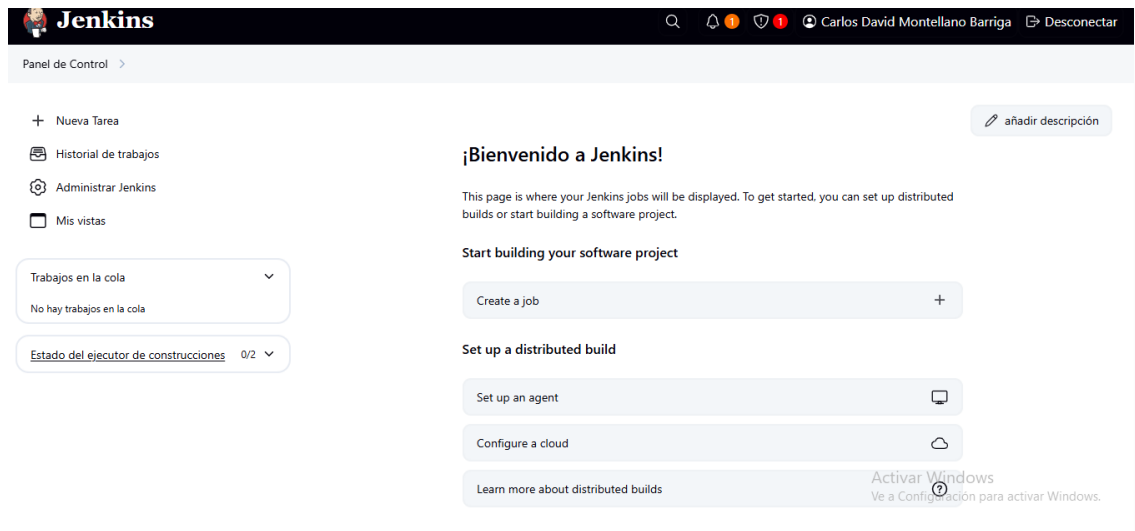
Nombre completo

Jenkins 2.504.1

Skip and continue as admin

Activar Windows
Ver a Configuración para...

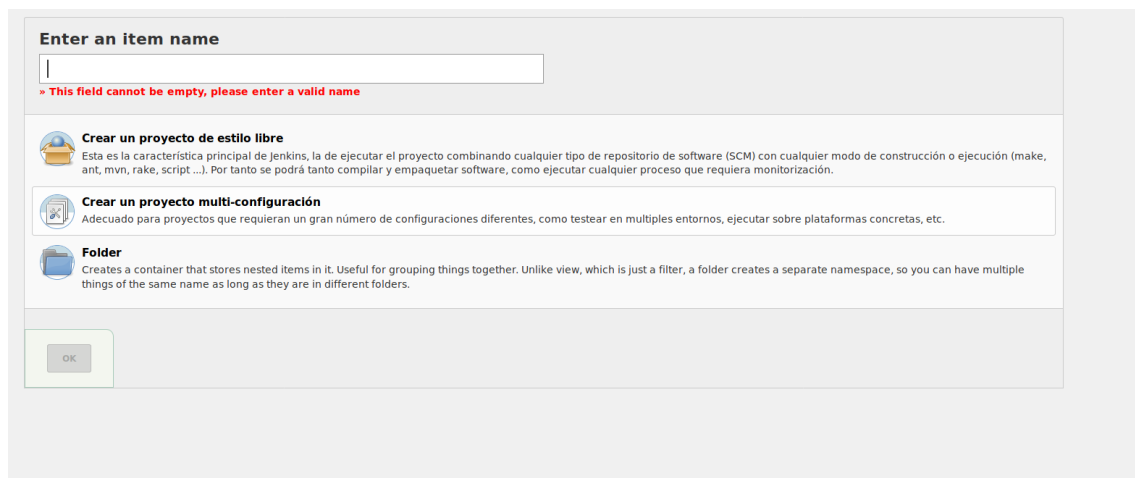
Save and Continue



Primeros pasos con Jenkins: Tutorial para empezar a usarlo

Crear una tarea

Jenkins se basa en tareas. Como su nombre indica una tarea es un trabajo o un conjunto de instrucciones que podemos programar para que ocurran con una determinada acción. Si pinchas en crear una nueva tarea:



Colocar el nombre del proyecto: Factorial

Coloca el siguiente código en el pipeline

```
pipeline {
  agent any

  tools {
    maven 'Maven 3.8.5' // Cambia por el nombre que hayas configurado en
    Jenkins
    jdk 'Java 11' // Cambia por el nombre que uses
  }

  stages {
```

```

stage('Clonar') {
    steps {
        git 'https://github.com/tu-usuario/FactorialApp.git' // Cambia por tu repo
    }
}

stage('Compilar') {
    steps {
        sh 'mvn clean compile'
    }
}

stage('Pruebas') {
    steps {
        sh 'mvn test'
    }
}

stage('Empaquetar') {
    steps {
        sh 'mvn package'
    }
}

post {
    success {
        echo ' Build y pruebas exitosas'
    }
    failure {
        echo ' Falló el proceso'
    }
}
}

```

Crea un repositorio en github FactorialApp

Clona el repositorio en local y escribe el codigo

Codigo en java para factorial con Maven
package com.example;

```

public class Factorial {

    public static long calcular(int numero) {
        if (numero < 0) {
            throw new IllegalArgumentException("El número debe ser >= 0");
        }
        long resultado = 1;
    }
}

```

```

        for (int i = 2; i <= numero; i++) {
            resultado *= i;
        }
        return resultado;
    }
}

```

Pruebas

```
package com.example;
```

```
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;
```

```
class FactorialTest {
```

```
    @Test
    void testFactorialDeCero() {
        assertEquals(1, Factorial.calcular(0));
    }

```

```
    @Test
    void testFactorialDeCinco() {
        assertEquals(120, Factorial.calcular(5));
    }

```

```
    @Test
    void testFactorialNegativo() {
        assertThrows(IllegalArgumentException.class, () -> Factorial.calcular(-1));
    }
}

```

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
        http://maven.apache.org/xsd/maven-4.0.0.xsd">

```

```

    <modelVersion>4.0.0</modelVersion>
    <groupId>com.example</groupId>
    <artifactId>factorial-app</artifactId>
    <version>1.0-SNAPSHOT</version>

```

```

    <properties>
        <maven.compiler.source>11</maven.compiler.source>
        <maven.compiler.target>11</maven.compiler.target>
        <junit.version>5.9.2</junit.version>
    </properties>

```

```
<dependencies>
  <!-- JUnit 5 -->
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter</artifactId>
    <version>${junit.version}</version>
    <scope>test</scope>
  </dependency>
</dependencies>

<build>
  <plugins>
    <!-- Compilar y ejecutar pruebas -->
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>3.0.0-M7</version>
    </plugin>
  </plugins>
</build>
</project>
```

Sube al repositorio

Haz correr en Jenkins el pipeline creado