Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.Texto

El contenido generado por IA puede ser incorrecto.Texto

El contenido generado por IA puede ser incorrecto.Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

cleanTexto

El contenido generado por IA puede ser incorrecto.

compile

Texto

El contenido generado por IA puede ser incorrecto.

Test

Texto

El contenido generado por IA puede ser incorrecto.Texto

El contenido generado por IA puede ser incorrecto.

Package

Texto

El contenido generado por IA puede ser incorrecto.

Install

Texto

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

**Paso 4: Manejo de dependencias, repositorios y estructura**

* Expliquen cómo se descargaron las dependencias.
  + El proyecto Maven lee el archivo pom.xml para identificar las dependencias que se definen.
  + Para cumplir con lo anterior, primero busca en el repositorio local que se encuentra en la carpeta oculta /.m2/repository.
  + En el archivo pom se le atribuyen alcances a las dependencias, como provided, runtime, test, las cuales definen si se usan solo en compilación, empaquetación, en producción, etc.
* Localicen el **repositorio local de Maven** y detallen qué tipo de archivos contiene.
* Muestren y expliquen la estructura del proyecto generada por Maven (src/main/java, src/main/resources, etc.).
  + Texto

    El contenido generado por IA puede ser incorrecto.
  + src/main/java: contiene M6Ae1GrupalApplication, ServletInitializer (porque empaquetamos WAR), los package de controller y model donde están el FormController, y PersonForm.
  + src/main/resources/templates: Thymeleaf (nuestro motor de vistas, no ocupamos jsp).
  + src/main/resources/static: archivos estáticos (CSS/JS/imagenes) si los tuviera, pero en esta ocasión no tengo nada.
  + target/: resultado del build (classes/ compilados, reportes de test y el .war).

**Paso 5: Documentación y presentación**

* Qué configuraron en el pom.xml
* Cómo resolvieron las validaciones
* Qué aprendieron sobre el ciclo de vida de Maven y los comandos utilizados
  + Ciclo de vida de Maven y comandos
    - mvn clean: limpia target/.
    - mvn compile: compila fuentes.
    - mvn test: ejecuta tests (reportes en target/surefire-reports).
    - mvn package: genera .war en target/.
    - mvn install: instala el artefacto en /.m2/repository.
  + Cómo ejecutar
    - Build: mvn clean package -DskipTests
    - Despliegue: copiar el WAR a tomcat/webapps/ o agregar el proyecto al servidor en STS.
    - URL: http://localhost:8080/M6\_AE1\_GRUPAL/form