

Paso 1: Creación del proyecto base con Maven en STS

Escogimos usar SpringBoot por la facilidad que permite crear el proyecto, configurarlo con la versión de Snapshot, el group y el artifact, al igual que incluir las dependencias a usar.

Pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.5.6</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.software</groupId>
  <artifactId>demo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>demo</name>
  <description>Demo project for Spring Boot</description>
  <url/>
  <licenses>
    <license/>
  </licenses>
  <developers>
    <developer/>
  </developers>
  <scm>
    <connection/>
    <developerConnection/>
    <tag/>
    <url/>
  </scm>
  <properties>
    <java.version>17</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-thymeleaf</artifactId>
```

```
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-validation</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-devtools</artifactId>
  <scope>runtime</scope>
  <optional>true</optional>
</dependency>
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <optional>true</optional>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>
</dependencies>

<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <configuration>
        <annotationProcessorPaths>
          <path>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
          </path>
        </annotationProcessorPaths>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
```

```

    <configuration>
      <excludes>
        <exclude>
          <groupId>org.projectlombok</groupId>
          <artifactId>lombok</artifactId>
        </exclude>
      </excludes>
    </configuration>
  </plugin>
</plugins>
</build>

</project>

```

Paso 2: Simulación del flujo de desarrollo con Maven

- **Mvn clean:**

```

PS D:\Bootcamp Java\Modulo 6\M6_AE1_ABPRO> mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.software:demo >-----
[INFO] Building demo 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- clean:3.4.1:clean (default-clean) @ demo ---
[INFO] Deleting D:\Bootcamp Java\Modulo 6\M6_AE1_ABPRO\target
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  0.429 s
[INFO] Finished at: 2025-10-20T20:01:07-03:00
[INFO] -----
PS D:\Bootcamp Java\Modulo 6\M6_AE1_ABPRO>

```

El comando `mvn clean` se encarga de eliminar la carpeta `target`, no incluye el código ni dependencias, solo deja el código limpio para compilar desde cero nuevamente.

- **Mvn compile:**

```
PS D:\Bootcamp Java\Modulo 6\M6_AE1_ABPRO> mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.software:demo >-----
[INFO] Building demo 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ demo ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO] Copying 0 resource from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.14.0:compile (default-compile) @ demo ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 1 source file with javac [debug parameters release 17] to target/classes
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  2.048 s
[INFO] Finished at: 2025-10-20T20:12:08-03:00
[INFO] -----
```

La función de mvn compile es compilar el código y copia los recursos a la carpeta target/classes, no empaqueta.

- **Mvn test:**

```
PS D:\Bootcamp Java\Modulo 6\M6_AE1_ABPRO> mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.software:demo >-----
[INFO] Building demo 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ demo ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO] Copying 0 resource from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.14.0:compile (default-compile) @ demo ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ demo ---
[INFO] skip non existing resourceDirectory D:\Bootcamp Java\Modulo 6\M6_AE1_ABPRO\src\test\resources
[INFO]
[INFO] --- compiler:3.14.0:testCompile (default-testCompile) @ demo ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.5.4:test (default-test) @ demo ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnit4PlatformProvider
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.software.demo.DemoApplicationTests
20:14:22.767 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils -- Could not detect default configuration classes for test class [com.software.demo.DemoApplicationTests]: DemoApplicationTests
does not declare any static, non-private, non-final, nested classes annotated with @Configuration.
20:14:22.849 [main] INFO org.springframework.boot.test.context.SpringBootTest -- Found @SpringBootTestConfiguration com.software.demo.DemoApplication for test class com.software.demo.DemoApplicationTests
20:14:22.980 [main] INFO org.springframework.boot.devtools.restart.RestartApplicationListener -- Restart disabled due to context in which it is running
```


Se encarga de compilar, ejecutar pruebas y empaquetar en un JAR ejecutable listo para correr.

- **Mvn install:**

```
PS D:\Bootcamp Java\Modulo 0\M0_AE1_ABPRD> mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.software:demo >-----
[INFO] Building demo 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ demo ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO] Copying 0 resource from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.14.0:compile (default-compile) @ demo ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ demo ---
[INFO] skip non existing resourceDirectory D:\Bootcamp Java\Modulo 0\M0_AE1_ABPRD\src\test\resources
[INFO]
[INFO] --- compiler:3.14.0:testCompile (default-testCompile) @ demo ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.5.4:test (default-test) @ demo ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnit4PlatformProvider
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.software.demo.DemoApplicationTests
20:16:32.229 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils -- Could not detect default configuration classes for test class [com.software.demo.DemoApplicationTests]: DemoApplicationTests
does not declare any static, non-private, non-final, nested classes annotated with @Configuration.
20:16:32.314 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper -- Found @SpringBootConfiguration com.software.demo.DemoApplication for test class com.software.demo.DemoApplicationTests
20:16:32.459 [main] INFO org.springframework.boot.devtools.restart.RestartApplicationListener -- Restart disabled due to context in which it is running
```

```
:: Spring Boot ::
(v3.5.0)

2025-10-20T20:16:32.675-03:00 INFO 26124 --- [demo] [
dule 0\M0_AE1_ABPRD] main] com.software.demo.DemoApplicationTests : Starting DemoApplicationTests using Java 21.0.3 with PID 26124 (started by Leonar Contreras in D:\Bootcamp Java\Mo
2025-10-20T20:16:32.675-03:00 INFO 26124 --- [demo] [
dule 0\M0_AE1_ABPRD] main] com.software.demo.DemoApplicationTests : No active profile set, falling back to 1 default profile: "default"
2025-10-20T20:16:33.633-03:00 WARN 26124 --- [demo] [
dule 0\M0_AE1_ABPRD] main] ion$DefaultTemplateResolverConfiguration : Cannot find template location: classpath:/templates/ (please add some templates, check your Thymeleaf configuratio
n, or set spring.thymeleaf.check-template-location=false)
2025-10-20T20:16:33.720-03:00 INFO 26124 --- [demo] [
dule 0\M0_AE1_ABPRD] main] com.software.demo.DemoApplicationTests : Started DemoApplicationTests in 1.269 seconds (process running for 2.021)
Mockito is currently self-attaching to enable the inline-mock-maker. This will no longer work in future releases of the JDK. Please add Mockito as an agent to your build as described in Mockito's documentation: https://javadoc.io/doc/org.mockito/mockito-core/latest/org.mockito/org.mockito/Mockito.html#0.3
Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
WARNING: A Java agent has been loaded dynamically (C:\Users\Leonar Contreras\.m2\repository\net\bytebuddy\byte-buddy-agent\1.17.7\byte-buddy-agent-1.17.7.jar)
WARNING: If a serviceability tool is in use, please run with -XX:-EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.212 s -- in com.software.demo.DemoApplicationTests
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.2:jar (default-jar) @ demo ---
[INFO]
[INFO] --- spring-boot:3.5.4:repackage (repackage) @ demo ---
[INFO] Replacing main artifact D:\Bootcamp Java\Modulo 0\M0_AE1_ABPRD\target\demo-0.0.1-SNAPSHOT.jar with repackaged archive, adding nested dependencies in BOOT-INF/.
[INFO] The original artifact has been renamed to D:\Bootcamp Java\Modulo 0\M0_AE1_ABPRD\target\demo-0.0.1-SNAPSHOT.jar.original
[INFO]
[INFO] --- install:3.1.4:install (default-install) @ demo ---
[INFO] Installing D:\Bootcamp Java\Modulo 0\M0_AE1_ABPRD\pom.xml to C:\Users\Leonar Contreras\.m2\repository\com\software\demo\0.0.1-SNAPSHOT\demo-0.0.1-SNAPSHOT.pom
[INFO] Installing D:\Bootcamp Java\Modulo 0\M0_AE1_ABPRD\target\demo-0.0.1-SNAPSHOT.jar to C:\Users\Leonar Contreras\.m2\repository\com\software\demo\0.0.1-SNAPSHOT\demo-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 4.994 s
[INFO] Finished at: 2025-10-20T20:16:35-03:00
[INFO] -----
```

El comando mvn install compila, prueba, empaqueta el JAR ejecutable y lo registra en tu repositorio local para reutilizarlo como dependencia.

Paso 3: Integración de un formulario con validación básica

Formulario

Nombre:

El nombre es obligatorio

Edad:

La edad es obligatoria


Formulario

Nombre:

El nombre es obligatorio

Edad:

La edad es obligatoria

 Value must be greater than or equal to 0.

Datos recibidos correctamente

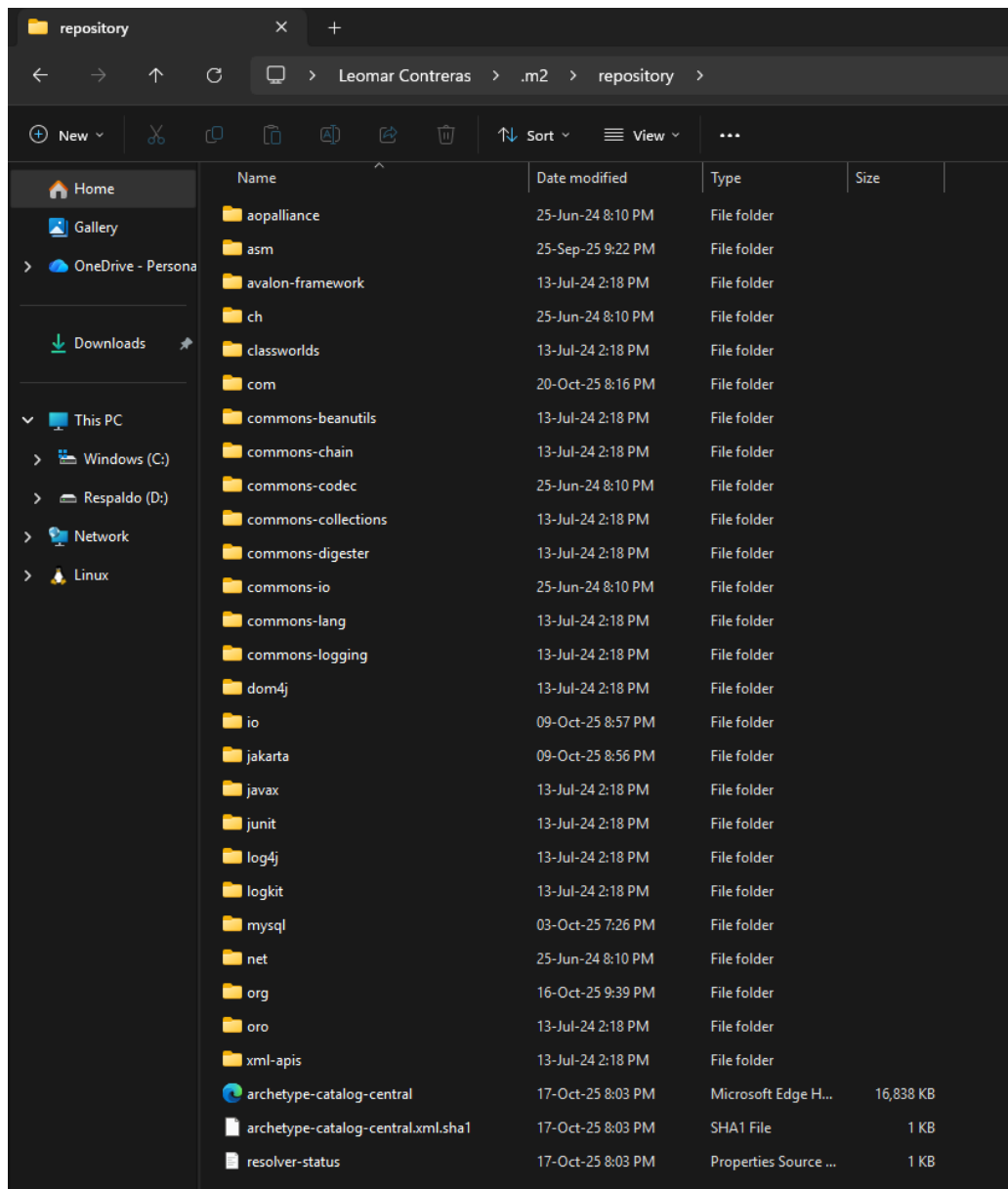
Nombre: Leomar

Edad: 35

[Volver al formulario](#)

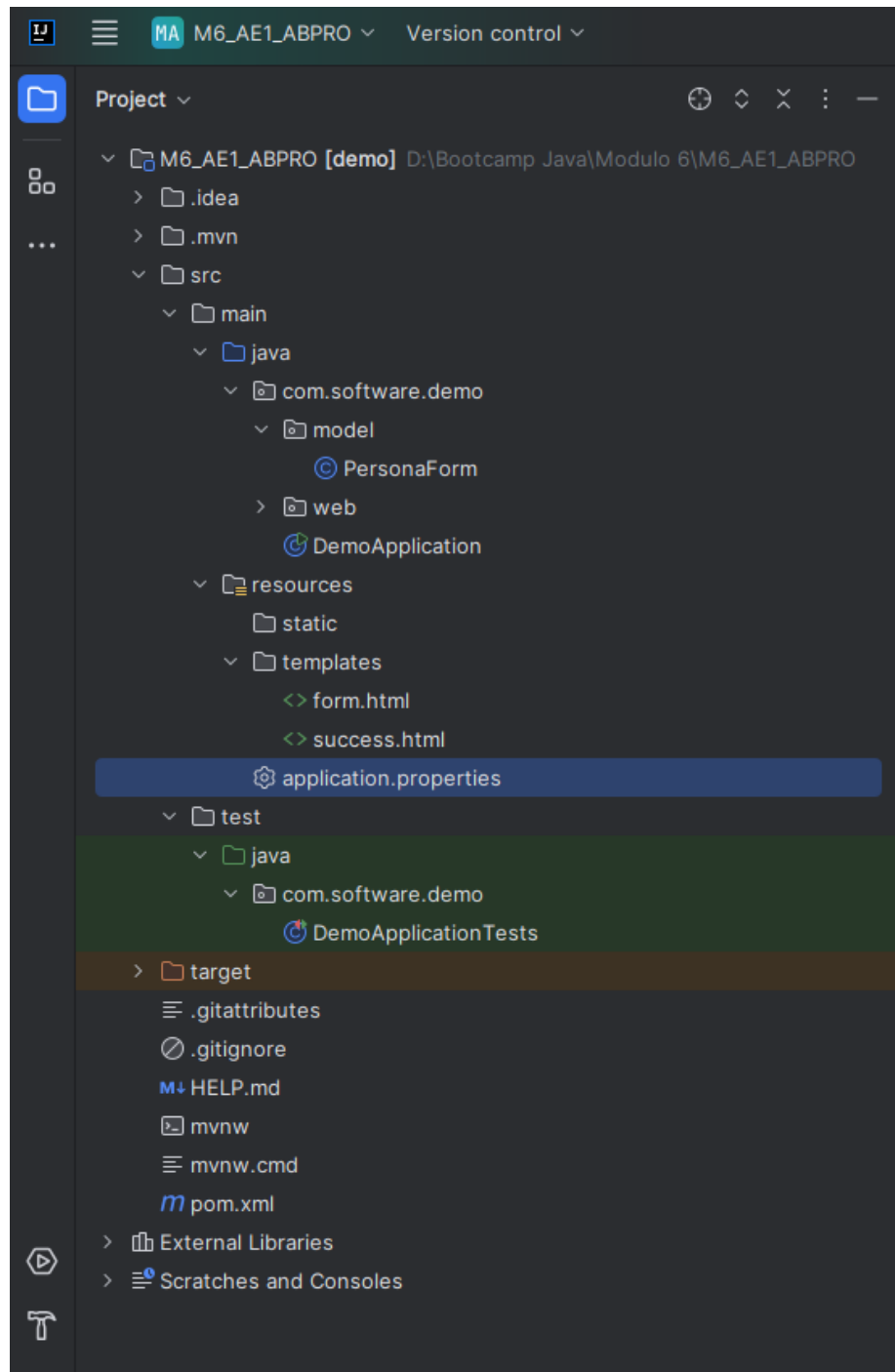
Paso 4: Manejo de dependencias, repositorios y estructura

- Como se descargaron las dependencias: al ejecutar el comando `mvn compile/test/package/install`, Maven lee el archivo `pom.xml` y usando el `parent` resuelve versiones anteriores.
 - Primero busca en el repositorio local, si las tiene las usa, de lo contrario lo descarga de Maven Central y procede a guardarlos en el repositorio local
- Repositorio local de Maven:



Dentro del repositorio local de Maven se encuentran todas las dependencias que han sido descargadas y usadas en diferentes proyectos.

- Estructura del proyecto generada por Maven:



Dentro del directorio Java se encuentran todos los paquetes relacionados con la aplicación, las clases principales, los controladores, los modelos, la conexión a la base de datos, entre otros. También los templates de los HTML y/o JSP, el paquete de los archivos Statics, de pruebas. El archivo pom.xml que alberga toda la configuración del proyecto.

Paso 5: Documentación y presentación

Como se resolvieron las validaciones:

Se resolvieron de la siguiente manera, en el modelo "PersonaForm", se colocó lo siguiente:

```
public class PersonaForm {  
  
    @NotBlank(message = "El nombre es obligatorio")  
    private String nombre;  
  
    @NotNull(message = "La edad es obligatoria")  
    @Min(value = 0, message = "La edad debe ser mayor o igual a 0")  
    @Max(value = 120, message = "La edad debe ser menor o igual a 120")  
    private Integer edad;  
}
```

De esta manera nos aseguramos de que se deba escribir un nombre y la edad debe ser mayor o igual a 0 y menor o igual a 120 años.

De los comandos de Maven utilizados aprendimos a como limpiar un proyecto, como eliminar el contenido de la carpeta target, como empaquetar el proyecto en un archivo JAR.