

ACTIVIDAD PRÁCTICA – GESTIÓN DE PROYECTOS

Por Pedro Berrueco

Trabajo propuesto como
cumplimiento de los requisitos para
la actividad práctica de la unidad 5.

Módulo de Entornos de Desarrollo.

Curso en Desarrollo de Aplicaciones
Multiplataforma Online.

Marzo 2023

Índice

ACTIVIDAD PRÁCTICA – GESTIÓN DE PROYECTOS.....	- 1 -
Descripción de la actividad Práctica	- 3 -
Introducción.....	- 3 -
Ejercicio 1.....	- 4 -
Ejercicio 2.....	- 5 -
Ejercicio 3.....	- 6 -
Ejercicio 4.....	- 7 -
Ejercicio 5.....	- 9 -
Ejercicio 6.....	- 10 -
Ejercicio 7.....	- 10 -
Ejercicio 8.....	- 12 -
Conclusiones.....	- 13 -

Descripción de la actividad Práctica

En esta actividad se debe crear un documento profesional tal y como se describe en el entregable. En cada apartado se debe incluir capturas de pantalla de los pasos que se han seguido, junto con la explicación de los pasos que se han dado. Lo que debes hacer es muy parecido a lo que has aprendido en la parte de teoría, por lo que te recomiendo que tengas cerca los contenidos de la unidad. Vamos a crear dos proyectos Maven en Eclipse “Paciente” y “Hospital”. Cada proyecto va a tener dentro una clase con el mismo nombre que el proyecto. La clase Hospital va a crear dos Pacientes diferentes.

Introducción

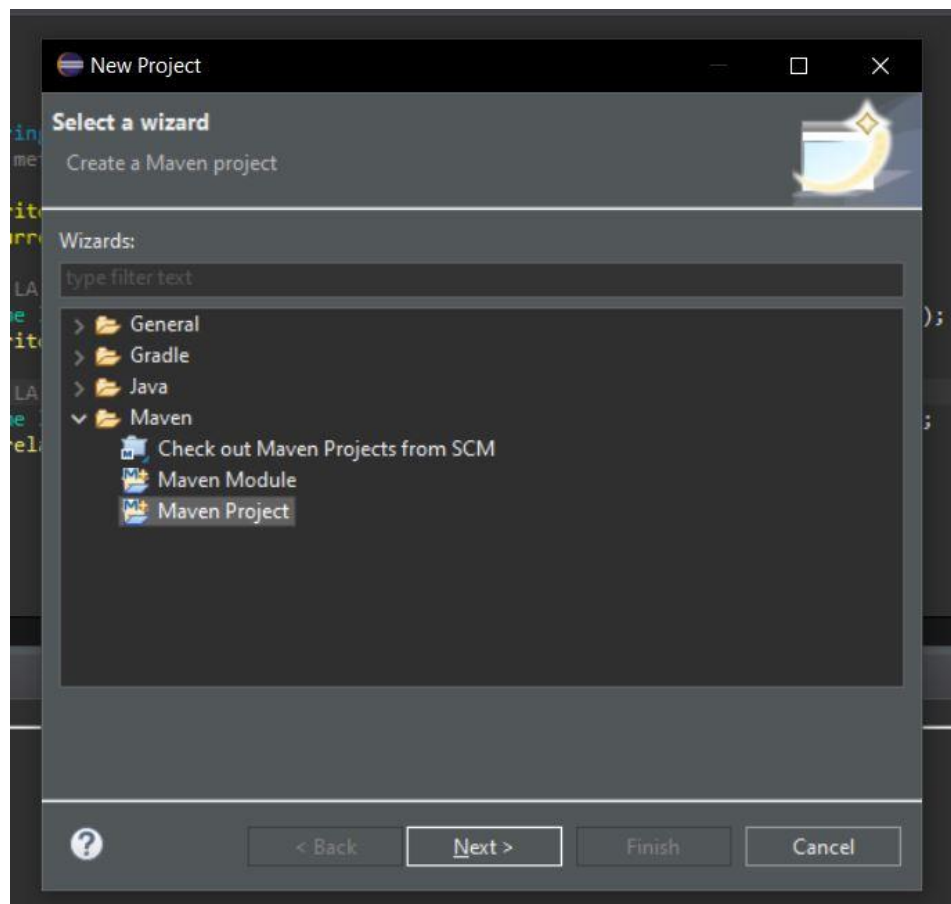
El objetivo de esta práctica es conocer si el alumno ha adquirido los conocimientos necesarios para crear dos proyectos utilizando la herramienta Maven. Con la resolución de los ejercicios se van trabajando los distintos apartados de la Unidad, y se realiza todo el camino que finaliza en la creación objetos de la clase “Paciente” en la clase “Hospital” alojada en otro proyecto. Para ello se debe haber definido una relación de confianza entra clases ayudándose de la aplicación de gestión de proyectos, Maven.

Ejercicio 1

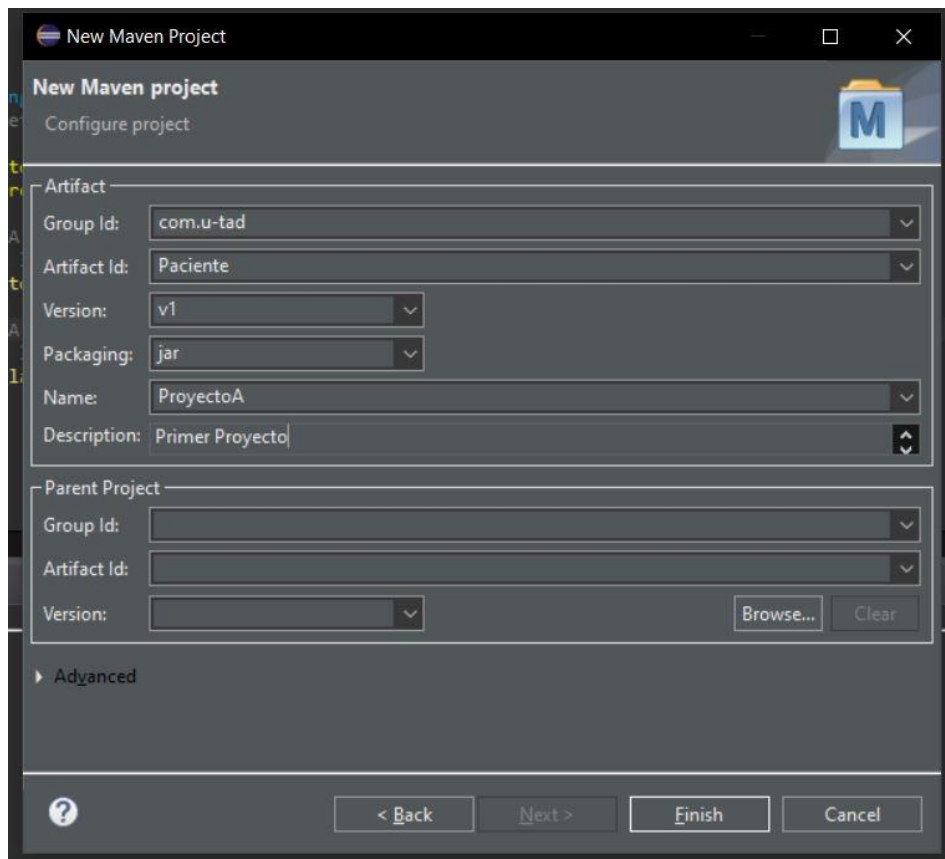
Proyecto paciente:

Desde Eclipse crea un proyecto Maven de tipo “simple”, con group-id “com.u-tad”, artifact id “Paciente”, versión “v1”, packaging “jar”, name “ProyectoA”, description “Primer proyecto”. Para ello seleccionamos la opción:

File > New > Project > Maven > Maven Project



Marcamos el check “Create a simple Project (skip archetype selection)”. Y cumplimentamos los campos como se puede apreciar en la imagen siguiente:



Ejercicio 2

Clase paciente:

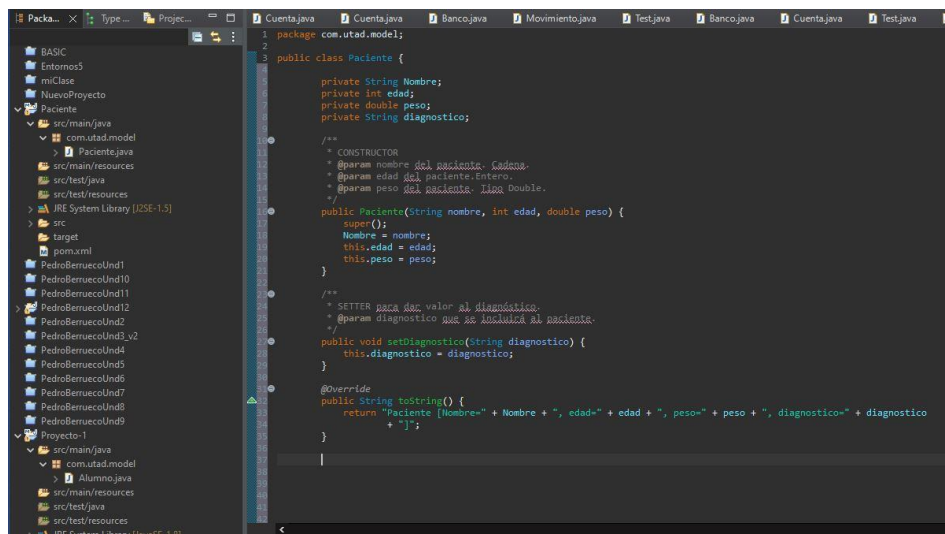
La clase debe estar en el paquete "com.utad.model" La clase debe tener 4 atributos privados:

- nombre: String
- edad: int
- peso: double
- diagnostico: String

También debe tener los siguientes métodos públicos:

- Constructor que sirva para inicializar nombre, edad y peso.
- Setter para dar valor al diagnóstico.
- toString para mostrar todos los atributos.

Se crea la clase “Paciente” siguiendo las indicaciones, dentro del paquete “com.utad.model” en la carpeta “src/main/java”.



Ejercicio 3

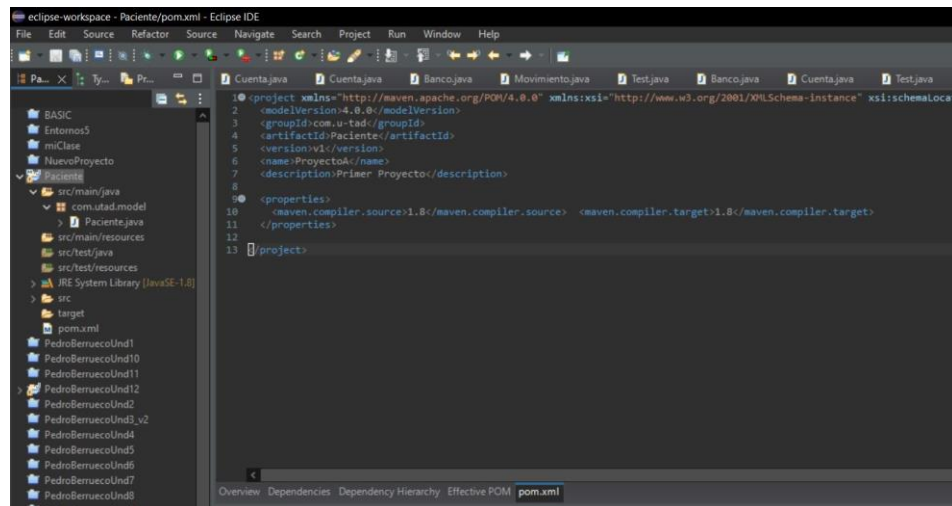
Versión 1.8:

Debes cambiar la versión del proyecto para que esté en 1.8:

Para ello hay que incluir el siguiente texto en la sección de “properties” en el fichero POM:

```
<properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
</properties>
```

Una vez guardados los cambios en el fichero pom.xml, realizamos click derecho sobre el nombre del y seleccionamos “Maven > Update Project”.

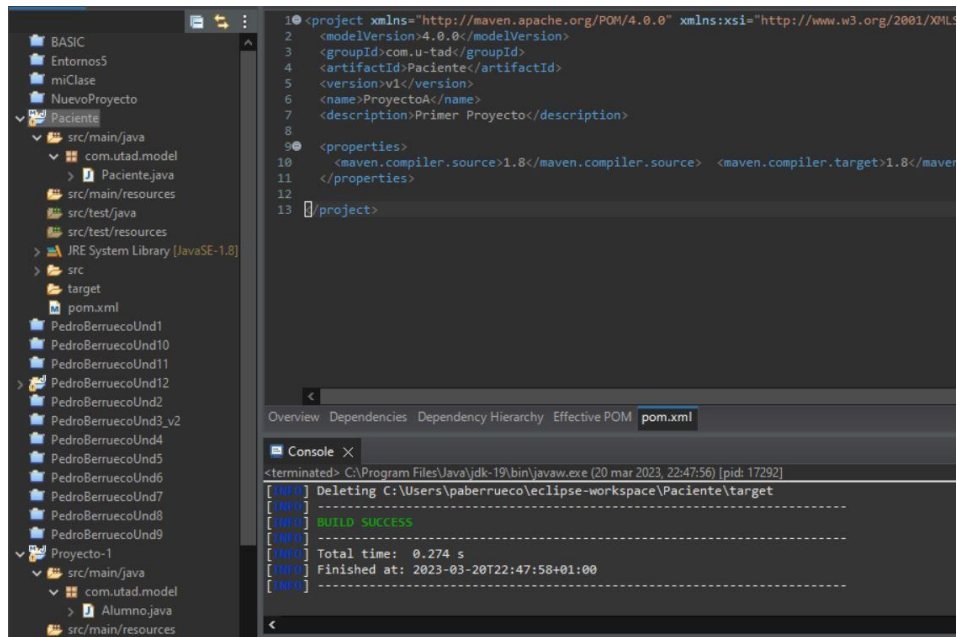


Ejercicio 4

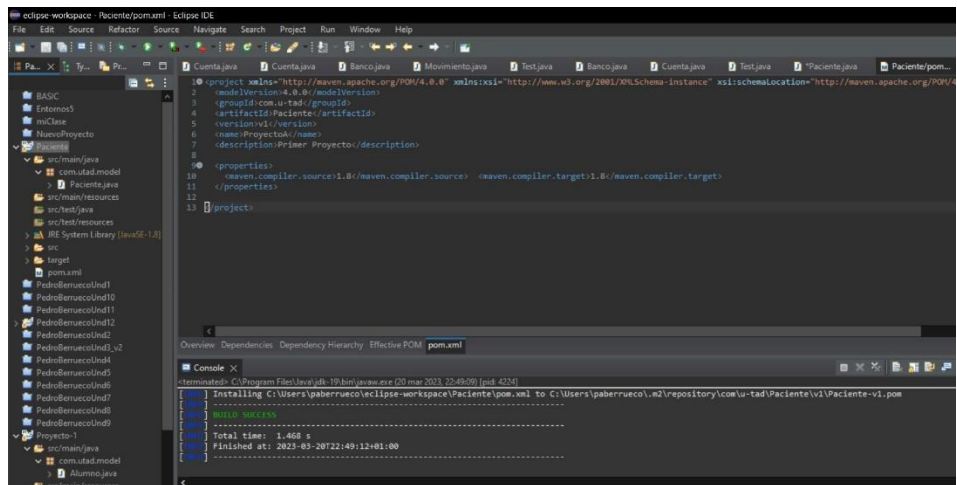
Install:

Debes generar el proyecto Paciente en local para que pueda ser posteriormente utilizado en el otro proyecto:

Nos aseguramos de que la carpeta target del proyecto esté vacía, para eso ejecutamos el comando clean, usando “Botón derecho sobre el proyecto Paciente > Run As > Maven clean”.



Ahora ya podemos crear el proyecto en local usando el comando install. “Botón derecho sobre el proyecto Paciente > Run As >Maven install”.



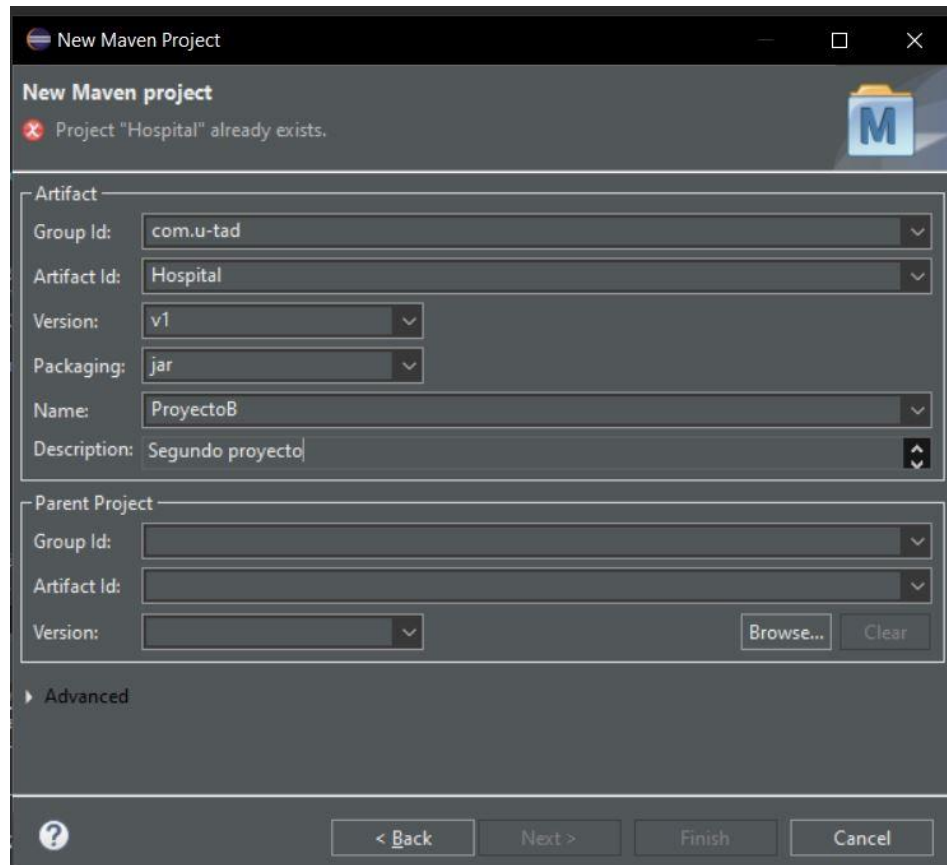
Ejercicio 5

Proyecto hospital:

Desde Eclipse crea otro proyecto Maven de tipo “simple”, con group-id “com.u-tad”, artifact id “Paciente”, versión “v1”, packaging “jar”, name “ProyectoB”, description “Segundo proyecto”. Para ello seleccionamos la opción:

File > New > Project > Maven > Maven Project > Marcamos el check “Create a simple Project (skip archetype selection)”. Y cumplimentamos los campos como se puede apreciar en la imagen siguiente:

*El enunciado nos pide volver a crear el artefacto “Paciente”, entendiendo que es una errata, lo he denominado “Hospital”.



The screenshot shows the 'New Maven Project' dialog box in Eclipse. At the top, there is a message: 'Project "Hospital" already exists.' with a red 'x' icon. Below this, the 'Artifact' section contains the following fields: Group Id: 'com.u-tad', Artifact Id: 'Hospital', Version: 'v1', Packaging: 'jar', Name: 'ProyectoB', and Description: 'Segundo proyecto'. The 'Parent Project' section has empty fields for Group Id, Artifact Id, and Version, along with 'Browse...' and 'Clear' buttons. The 'Advanced' section is collapsed. At the bottom, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

Ejercicio 6

Versión 17:

Debes cambiar la versión del proyecto para que esté en 17:

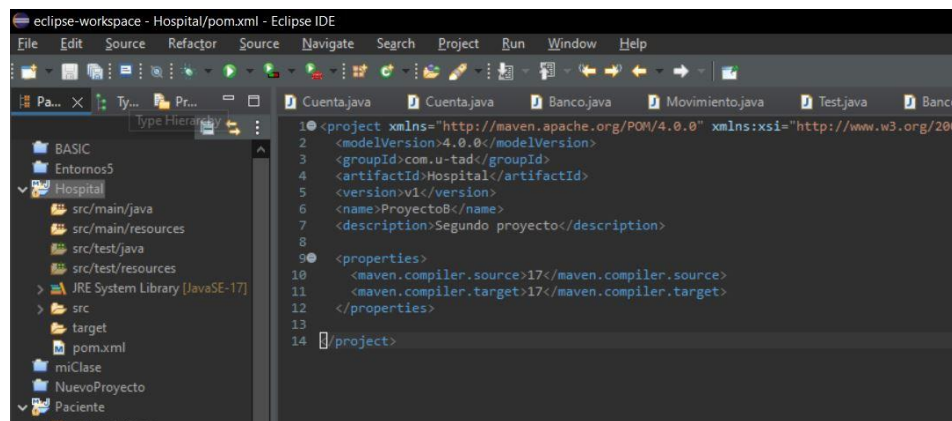
Para ello hay que incluir el siguiente texto en la sección de “properties” en el fichero POM:

```
<properties>

    <maven.compiler.source>17</maven.compiler.source>

    <maven.compiler.target>17</maven.compiler.target>

</properties>
```



Una vez guardados los cambios en el fichero pom.xml, realizamos click derecho sobre el nombre del y seleccionamos “Maven > Update Project”.

Ejercicio 7

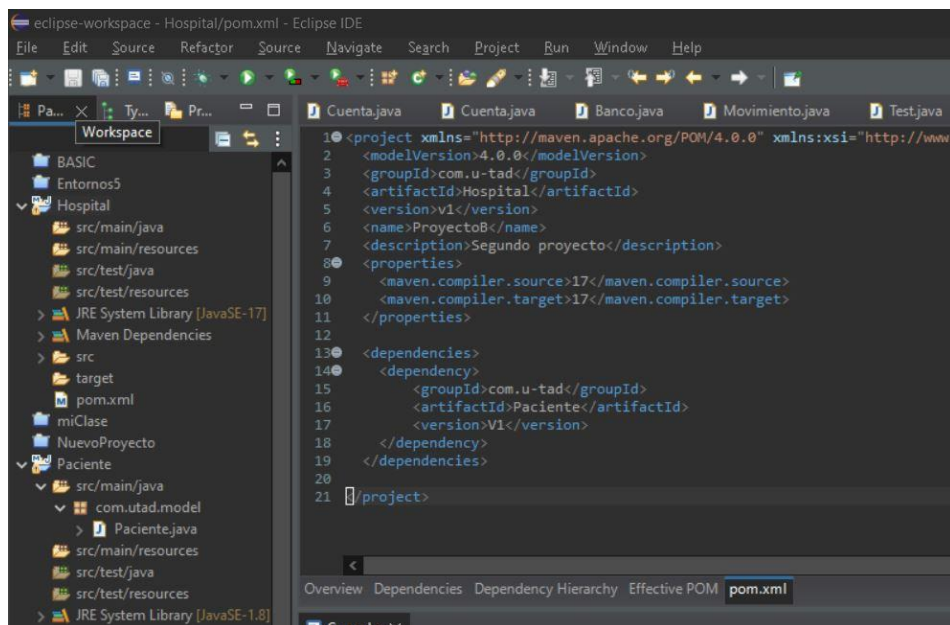
Dependencias:

Debes hacer todo lo necesario para que desde este segundo proyecto se pueda utilizar la clase Paciente que has creado anteriormente:

Debemos actualizar las dependencias del proyecto “Hospital”, para que pueda utilizar las clases del proyecto Paciente.

```
<dependencies>
  <dependency>
    <groupId>com.u-tad</groupId>
    <artifactId>Paciente</artifactId>
    <version>V1</version>
  </dependency>
</dependencies>
```

Para ello, hay que copiar este código en el fichero POM.xml del proyecto “Hospital”, como se ve en la siguiente imagen:



Guardamos el fichero y pulsamos “Click derecho > Maven > Update Project”.

Ejercicio 8

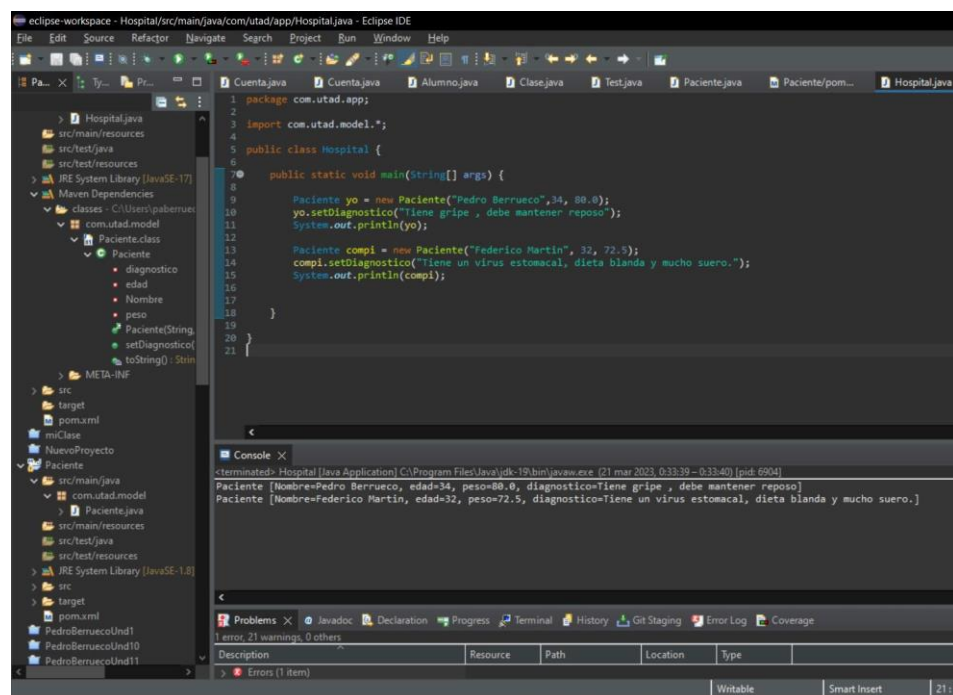
Clase hospital:

La clase debe estar en el paquete “com.utad.app”. La clase debe tener un “main” en el que debes crear dos objetos de la clase paciente:

- Un primer objeto con tu nombre y tus datos. Invéntate un diagnóstico y muestra el objeto por consola.
- Un segundo objeto con el nombre de un compañero. De igual forma, invéntate un diagnóstico y muestra el objeto por consola:

Se crea la clase “Hospital” con un método “main”, siguiendo las indicaciones, dentro del paquete “com.utad.app” en la carpeta “src/main/java”.

Para poder usar los métodos de la clase Paciente se importa la clase generada anteriormente con la opción “install”:



The screenshot shows the Eclipse IDE interface. The left sidebar displays the project structure for 'Hospital', including the 'src/main/java' directory where 'Hospital.java' is located. The main editor window shows the code for 'Hospital.java', which imports 'com.utad.model.*' and defines a 'main' method. The 'main' method creates two 'Paciente' objects: 'yo' (Pedro Berrueto, 34, 88.0) and 'compi' (Federico Martin, 32, 72.5), each with a specific diagnosis. The console at the bottom shows the output of the program, displaying the details of both patients. The 'Problems' view at the bottom indicates 1 error and 21 warnings.

```
1 package com.utad.app;
2
3 import com.utad.model.*;
4
5 public class Hospital {
6
7     public static void main(String[] args) {
8
9         Paciente yo = new Paciente("Pedro Berrueto", 34, 88.0);
10        yo.setDiagnostico("tiene gripe , debe mantener reposo");
11        System.out.println(yo);
12
13        Paciente compi = new Paciente("Federico Martin", 32, 72.5);
14        compi.setDiagnostico("tiene un virus estomacal, dieta blanda y mucho suero.");
15        System.out.println(compi);
16
17    }
18
19 }
20
21 }
```

Console Output:

```
terminated: Hospital [Java Application] C:\Program Files\Java\jdk-19\bin\java.exe (21 mar 2023, 0:33:39 - 0:33:40) [pid 6904]
Paciente [Nombre=Pedro Berrueto, edad=34, peso=88.0, diagnostico=tiene gripe , debe mantener reposo]
Paciente [Nombre=Federico Martin, edad=32, peso=72.5, diagnostico=tiene un virus estomacal, dieta blanda y mucho suero.]
```

Como vemos, la relación de dependencia se genera correctamente.

Conclusiones

Con la consecución de esta práctica se repasa la unidad en profundidad y se obliga al alumno a utilizar Maven, entender el motivo de su función y el alcance que esta herramienta ofrece como apoyo a la gestión de proyectos.

La práctica realmente es muy sencilla, cómo indica en el enunciado, si la sigues con la unidad abierta, es casi como un guía-burros del que seguir los pasos y van saliendo los ejercicios.

En cualquier caso, creo que el objetivo de la práctica está más que logrado y muy bien enfocado, pues después de terminarla tengo muy claro el funcionamiento de Maven y ya puedo empezar a crear todos mis proyectos con Maven sin ningún tipo de problema o duda que pudiera surgirme, la exportación de librerías me parece muy sencilla utilizando esta herramienta y en términos generales estoy muy satisfecho con la actividad práctica realizada, no se hace pesada, se termina en un tiempo relativamente rápido y pienso que favorece el aprendizaje y fija los conocimientos de la unidad.

En conclusión, esta práctica ha sido una excelente herramienta educativa para comprender los conceptos de las aplicaciones de Gestión de Proyectos. Hemos aprendido a crear proyectos, crear librerías, y utilizar en otros proyectos de forma eficiente, fácil y rápida. Esto nos ha permitido mejorar nuestras habilidades en el manejo de Eclipse y avanzar un poquito más hacia nuestra meta de ser buenos desarrolladores.