

Gestión de equipos de trabajo

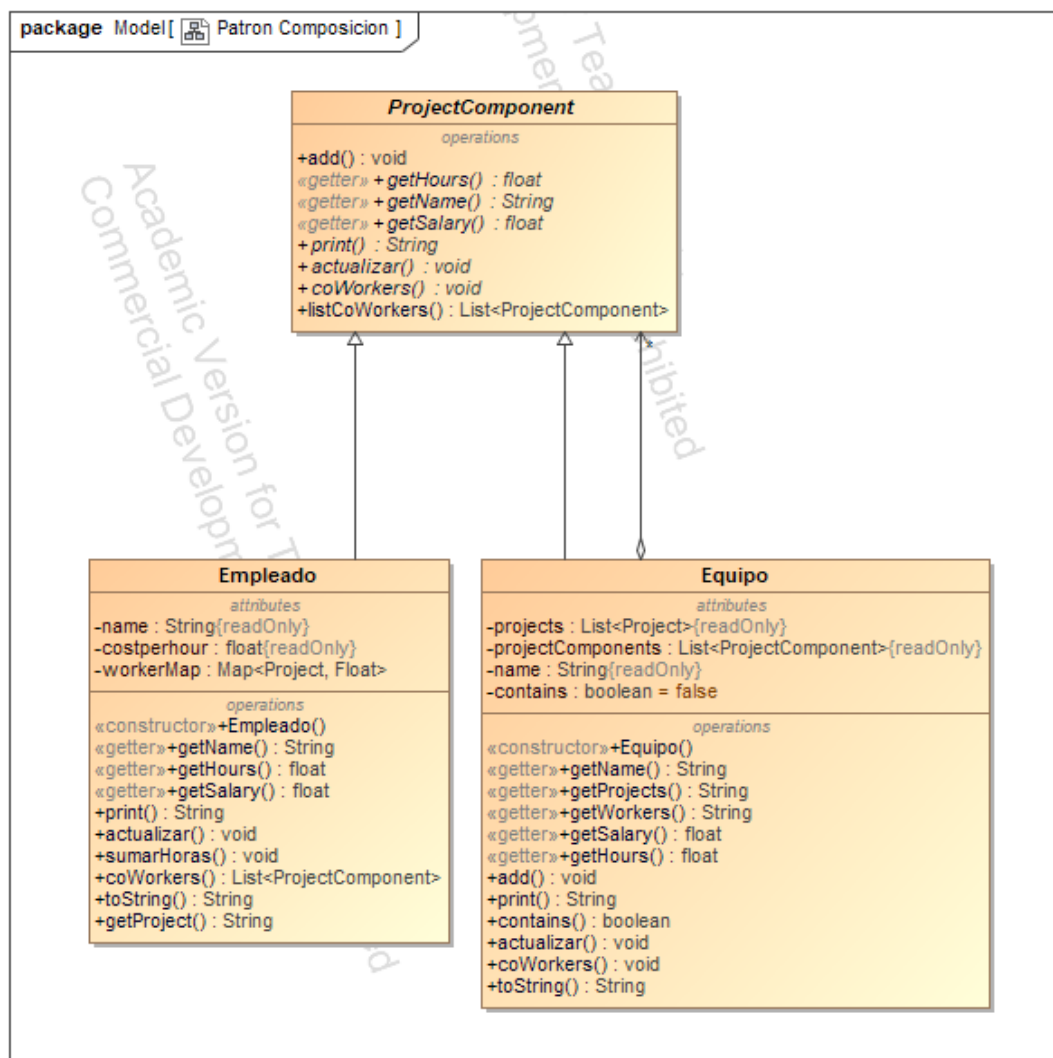
Patrones que usamos en esta práctica:

- **Patrón composición:**

Toda la jerarquía de este proyecto está basada en un patrón composición donde:

- Equipo juega el rol de **Composición**
- Empleado juega el rol de **Componente concreto**
- ProjectComponent juega el rol de **Componente**
- Project juega el rol de **Cliente**

Hemos decidido usar este patrón para este ejercicio porque nos facilita la introducción de nuevos componentes, además es la opción más rápida y correcta de tratar tanto a equipo como a empleado como componentes del propio proyecto.



Usamos la clase abstracta ProjectComponent para tratar a Equipo y a Empleado como componentes del proyecto.

Principios que usamos en esta práctica:

- Principio abierto-cerrado:

Partiendo de que Equipo y Empleado heredan de la clase abstracta ProjectComponent, aunque ambos hereden los métodos de la clase abstracta, cada uno los ejecuta de una forma distinta. Es decir, cambian los que hace una clase sin tocar el código de la clase. Un ejemplo claro es el Salary, que un empleado devuelve su salario pero un equipo hace un bucle buscando a todos los empleados que forman parte de ese equipo para devolver el total.

Diagrama de clases y secuencia: (Los de secuencia son 3 porque hemos representado el add (que actualiza a todos los componentes del equipo), los coworkers y el print.

