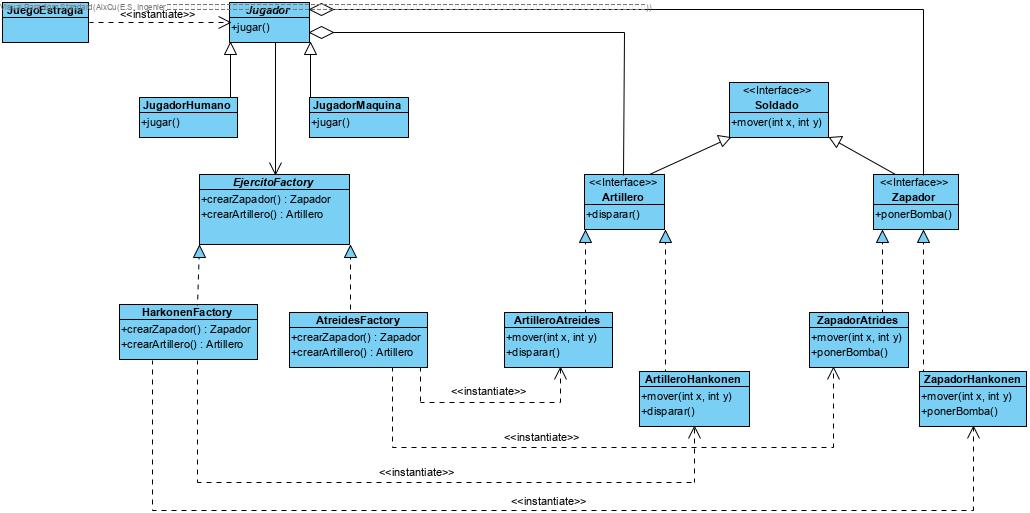
PARTE 1. GoF

DAGSS\_2

Currás Rodríguez, Alexandre

Vila Fernández, David

# Ejercicio 1



# Ejercicio 2

Utilizamos el patrón Builder debido a la necesidad de crear distintas representaciones de la colección de Libros. En este caso hay 2 representaciones distintas, una en memoria (clase Biblioteca) y otra en forma de documento XML generado por la clase XMLBibliotecaBuilder.

De este modo, la manera de crear estas 2 representaciones es independiente de las partes que las componen (Libros) y es posible añadir nuevas representaciones sin modificar el código existente.

# Ejercicio 3

Usamos el patrón Composite para representar una jerarquía de árbol, donde las hojas siempre son programas y las ramas son los trabajos que ejecutan o bien los programas u otros trabajos. También nos aporta una gran flexibilidad, permitiendo introducir nuevos programas sin modificar el resto de clases.

Además, usamos una modificación del patrón Builder, donde la clase TrabajoBuilder ordena la construcción del objeto Trabajo. De esta forma conseguimos crear un framework reutilizable, que usa la clase abstracta Trabajo de la que heredan TrabajoParalelo y TrabajoSerie.

Usando este patrón buscamos abstraer la creación de objetos complejos (Trabajo), de manera que en TrabajoBuilder centralizamos el proceso de creación de distintos tipos de trabajo.