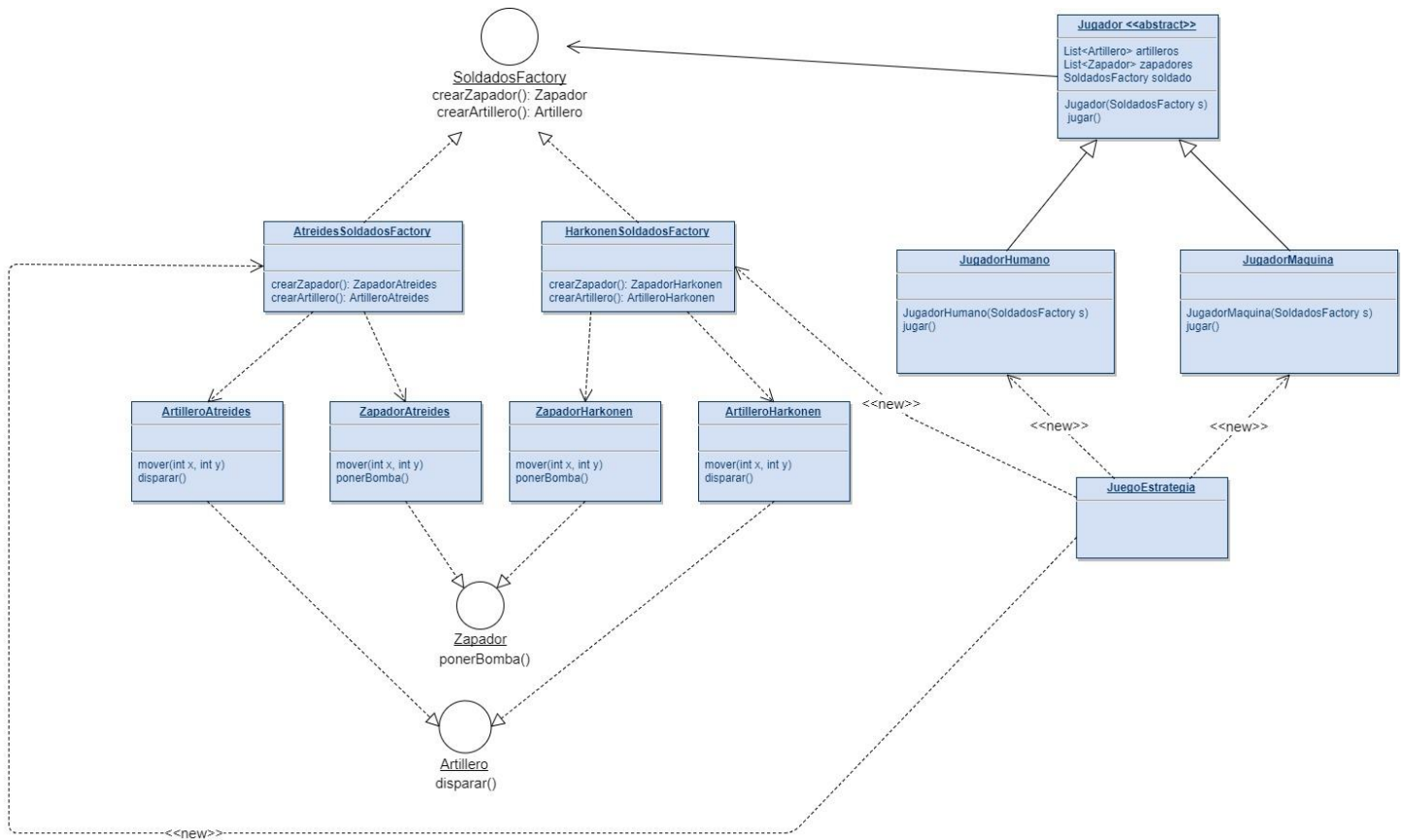


## PARTE 1. GoF-1

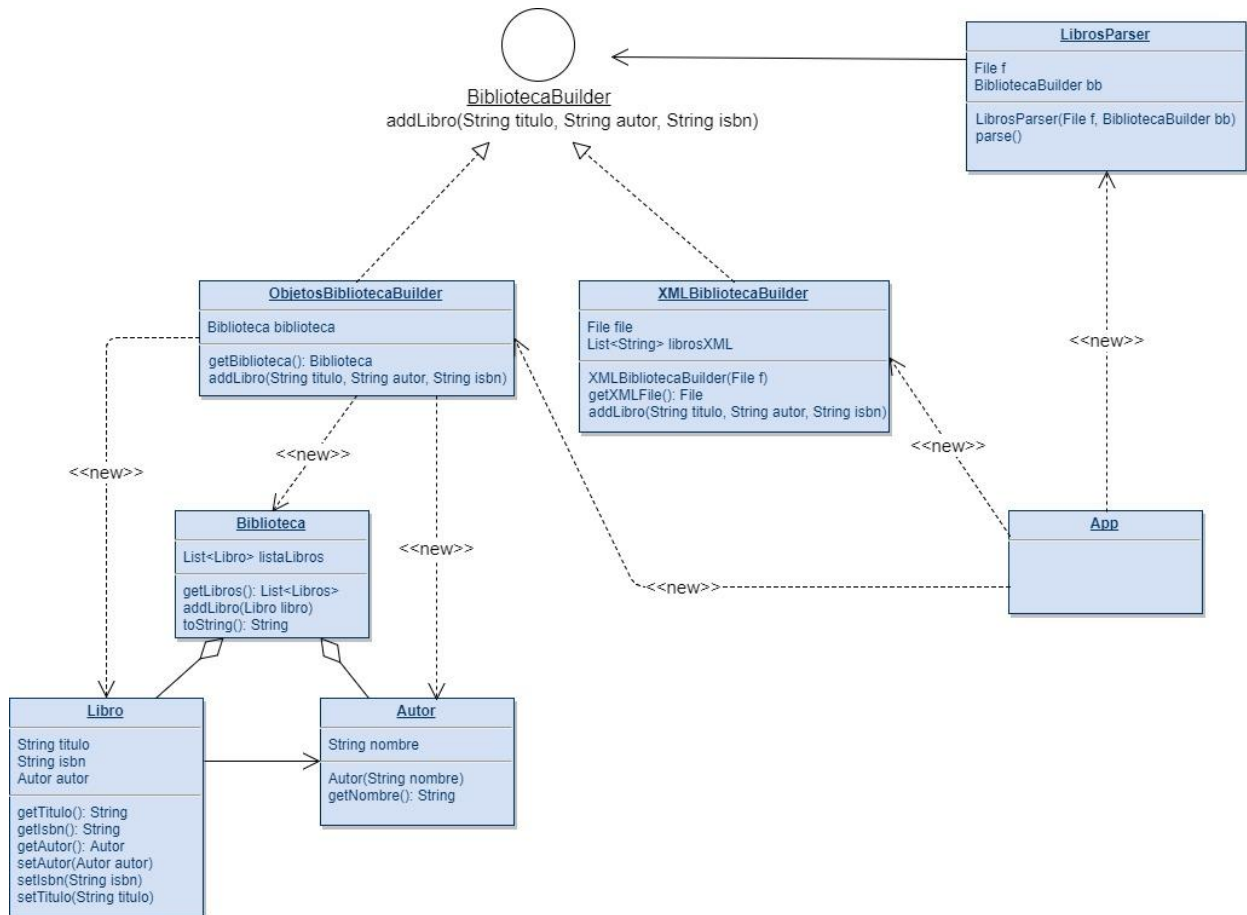
Realizado por: Brais Domínguez Álvarez y Silvia Rodríguez Iglesias.

### Ejercicio 1. SoldadosFactory.



### Ejercicio 2. Biblioteca.

Se ha empleado el patrón Builder a la hora de crear las diferentes representaciones de Biblioteca. Con la interfaz BibliotecaBuilder podemos implementar diferentes formas de representación (como XML o la estructura Biblioteca orientada a objetos en memoria) sin la necesidad de modificar el resto de código, como el encargado de tomar los datos de un fichero, y sin la necesidad de saber cómo deben construirse estos objetos. Esto permite además añadir otras representaciones en un futuro de manera muy sencilla. El patrón Builder, por lo tanto, nos permite crear objetos independientemente de las partes que lo componen y cómo estas se combinan. Su principal diferencia con AbstractFactory es que permite construir los objetos paso a paso.



### Ejercicio 3. Programas

Uno de los patrones empleados en este ejercicio es el patrón Builder, al igual que en el anterior. En este caso se emplea para la creación de diferentes tipos de trabajo (en serie o en paralelo). Además, podemos observar un patrón Bridge, ya que, para evitar tener los programas duplicados (en serie y en paralelo, es decir `ProgramASerie` y `ProgramAParalelo`) se abstraen y se crean sólo un trabajo en paralelo y otro en serie, evitando así muchas combinaciones. Así, se podría añadir otra forma de trabajo sin problema, la cual también ejecutaría cualquier tipo de programa sin necesidad de añadir nuevas implementaciones de dichos programas. Con este método se puede variar la implementación en tiempo de ejecución. Además, se consigue evitar una explosión de clases permitiendo que expandir el sistema sea mucho más sencillo.

