Resumo do Traballo de Fin de Grao

Introdución

AIBench é un *workbench* para intelixencia artificial desenvolvido polo grupo SING. Este *workbench* está especialmente deseñado para traballar con modelos de aplicación IPO (Input-Process-Output), nos que existen unha serie de operacións que reciben unhas entradas e xeran unhas saídas, as cales poden ser encadeadas entre si. Estas operacións defínense en *plugins* que, ademais, tamén permiten engadir tipos de datos e interfaces gráficas personalizadas. Cada dato do *workbench*, tanto de entrada como de saída, pode verse como un *viewer*. O usuario interactúa coa aplicación solicitando execucións de procesos e explorando os tipos de datos a través dos *viewers*.

Un xerador de código é unha ferramenta que permite construír proxectos completos ou partes útiles destes de maneira automática. O emprego deste tipo de ferramentas evita realizar parte do código manualmente, conseguindo un aforro de tempo no desenvolvemento de proxectos. Algúns dos máis coñecidos son:

1. Bake para CakePHP: consola que pode xerar elementos de CakePHP como modelos, vistas e controladores.
2. Yeoman: ferramenta que contén diferentes tipos de xeradores de código, facilitándolle ao usuario a construción do esqueleto da aplicación a crear.
3. @angular/cli: ferramenta para a construción de aplicacións Angular.

Neste traballo fin de grao proponse o deseño e implementación dun xerador de código para AIBench que axude aos programadores na creación das diferentes partes que compoñen dita aplicación.

Obxectivos

O obxectivo principal deste proxecto é desenvolver unha ferramenta de asistencia para creación de *plugins* para AIBench. Esta ferramenta debe permitir:

- Creación de Operacións (*Operations*). Definición dunha clase Java onde cada porto (punto onde algúns datos poden recibirse, producirse ou ambos) está asociado a un método.
- Creación de Tipos de Dato (*Datatypes*). Definición dunha clase Java empregada como entrada e saída das operacións.
- Creación de Vistas (*Views*). Definición de clases que herdan de `JComponent` e que se empregan para mostrar os tipos de datos dentro do *workbench*.
- Personalización dos Diálogos de Entrada (*Input Dialogs*). Modificación visual dos diálogos xerados por defecto, podendo crearse dende cero ou ampliando unha clase xa existente.
- Inclusión de Axudas (*Helps*). Engadir os arquivos JavaHelp nunha carpeta global e asocialos coa operación ou vista correspondente.

Ademais, deberá actualizar os ficheiros de configuración do *plugin* sempre que sexa necesario, sendo o máis importante o ficheiro `plugin.xml`.

Descrición técnica

A realización deste proxecto levarase a cabo mediante o desenvolvemento dun *plugin* Maven que funcione por liña de comandos, cumprindo os obxectivos anteriormente descritos. Dito *plugin* deberá ser extensible para adaptarse ás novas funcionalidades de AIBench que se poidan engadir no futuro.

Na *Figura 1* pode apreciarse como, mediante o emprego do xerador, as operacións son creadas de forma allea ao usuario. No proceso de creación da operación defínese internamente a clase que representa á operación, a cal contén os métodos (operacións) que recibirán e/ou producirán datos (tipos de datos). Ademais, anótase e conéctase dita operación a AIBench a través do arquivo `plugin.xml`.

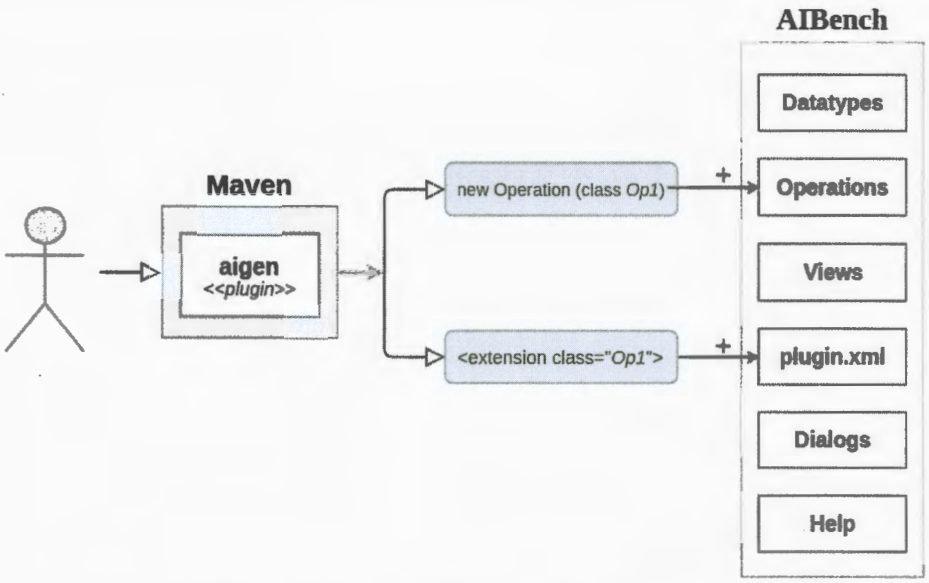

Figura 1. Creación dunha nova operación mediante aigen.

Método de Desenvolvemento

Este proxecto desenvolverase empregando unha metodoloxía áxil baseada en Scrum, pensada para adaptarse ás características especiais deste traballo, como o son o feito de que será desenvolvido por un equipo unipersoal, e que as persoas implicadas no proxecto son, unicamente, o propio alumno e o titor do mesmo, ademais de que o número de horas de dedicación é moi reducido (300 horas).

As modificacións e/ou adaptacións respecto á metodoloxía Scrum que se aplicarán son as seguintes:

- Farase uso das historias de usuario para a toma de requisitos.
- As reunións diarias (*daily meetings*) realizaranse cunha frecuencia semanal, participando o alumno e o titor.
- O alumno será o único membro do equipo de desenvolvemento.
- O alumno asumirá o rol de Product Owner e o titor asumirá o rol de Scrum Master.
- O alumno será incluído como un dos actores interesados no proxecto.