# Ejercicios de Auto-evaluación

En este documento se detallan las decisiones de diseño tomadas para la realización del proyecto de Ejercicios de Auto-evaluación, cuyos requisitos se recogen en el documento proporcionado en moodle[[1]](#footnote-1). Además se comentará el lenguaje de dominio específico desarrollado y su forma de uso.

## Decisiones de Diseño

Tras el análisis de los requisitos del proyecto decidimos realizar un diseño bottom-up. Comenzamos realizando un diseño inicial del meta-modelo junto con las restricciones que consideramos necesarias. A continuación, pasamos a realizar el diseño de la interfaz de la autoevaluación. Después de tener en cuenta la interfaz tuvimos que modificar el meta-modelo inicial, para que se ajustase a la interfaz y simplificamos las restricciones, dejando las complejas para el validador de xText. Quedando al final el siguiente esquema del meta-modelo:

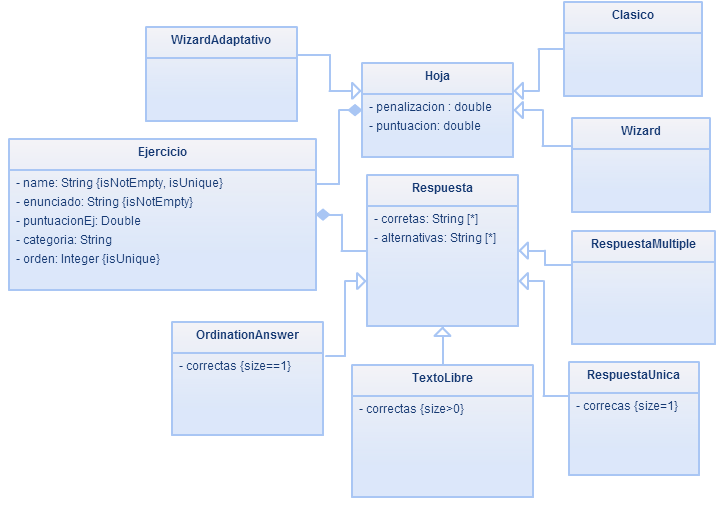


Figura Diagrama de clases del meta-modelo con restricciones

Decidimos crear una clase abstracta Hoja de la cual heredarían los distintos tipos de hojas: Clasica, Wizard y WizardAdaptativo. Esta clase contiene todas las variables necesarias para una hoja de ejercicios, la puntuación general que se asigna por defecto a todos los ejercicios, la penalización que se aplique y un array de ejercicios. Los hijos de la clase especifican el estilo de la hoja. Otra opción de diseño sería sustituir la herencia por una variable de estilo en la clase Hoja, pero decidimos que el diseño descrito anteriormente puesto que nos permite una mayor escalabilidad en futuros proyectos.

Un Ejercicio está compuesto por un nombre y un orden únicos, un enunciado, una categoría, una puntuación y una Respuesta. Cada tipo de respuesta única, múltiple, de texto libre y de ordenación heredan de la clase abstracta Respuesta, que contiene dos arrays de Strings de respuestas correctas o incorrectas. Cada tipo de respuesta tiene sus propias restricciones sobre la cantidad de respuestas correctas y usará las alternativas cuando sea necesario.

Escogimos implementar la interfaz gráfica con Java swing, puesto que ya teníamos más conocimientos sobre el mismo que de Ruby. En la siguiente Figura se puede observar el diagrama de clases de la interfaz gráfica, la cual se genera desde el lenguaje de dominio específico.

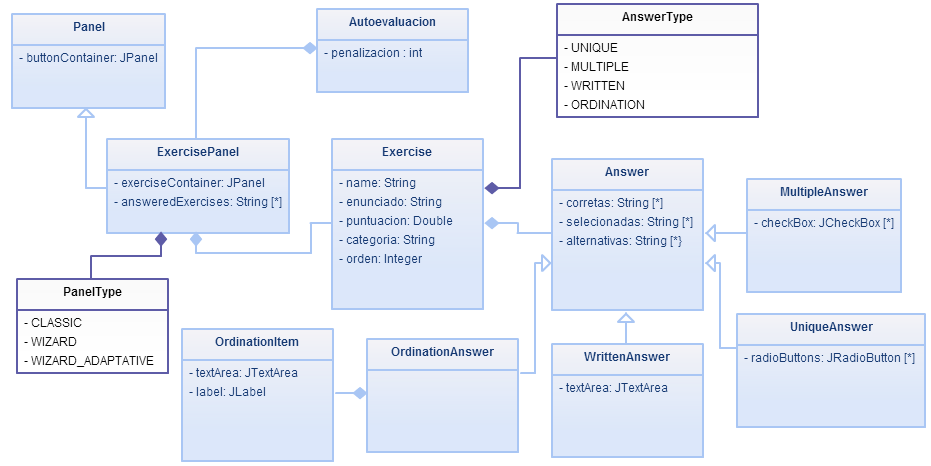


Figura Diagrama de clases de la interfaz

## Lenguaje de Dominio Específico Desarrollado

xText nos proporciona una gramática por defecto, en principio la mantenemos puesto que nos parece bastante intuitiva, al menos para usuarios con cierta experiencia en programación. Simplificamos esta gramática omitiendo redundancias y cambiando algunos de los términos claves, para facilitar al usuario la comprensión de la gramática. De modo que al final nos queda lo siguiente:

<Tipo\_de\_Hoja> **{**

**puntuacion** <puntuacion>

**penalizacion** <penalizacion>

**ejercicios{**

**Ejercicio** <nombre\_ejercicio> **{**

**enunciado** <enunciado>

**puntuacion** <puntuacion\_ejercicio>

**categoria** <categoria>

**orden** <orden>

<Tipo\_de\_respuesta> **{**

**correctas {**<correctas>**}**

**alternativas {**<alternativas>**}**

**}**

**},**

**…**

**}**

**}**

* ***<Tipo\_de\_Hoja>:*** indica el estilo de la hoja de ejercicios, puede ser Clasico, Wizard o WizardAdaptativo.
* ***<puntuacion>:*** es la puntuación que se asigna por defecto a todos los ejercicios. Es un double por lo que se ha de añadir “.0” si se quiere asignar un número entero. No se controlan límites pero se muestra un warning si se introduce un número negativo.
* ***<penalizacion>:*** es la penalización que se asigna a todos los ejercicios, en el caso de que se falle al responder. Es un double por lo que se ha de añadir “.0” si se quiere asignar un número entero. No se controlan límites pero se muestra un warning si se introduce un número negativo.
* ***<nombre\_ejercicio>:*** es un String que identifica el ejercicio por lo que debe ser único.
* ***<orden>:*** es un integer que indica el orden del ejercicio, también ha de ser único.
* ***<enunciado>:*** es un String con el enunciado del ejercicio.
* ***<puntuacion\_ejercicio>:*** es un double con la puntuación del ejercicio. Este campo no es obligatorio, en caso de escribir una puntuación al ejercicio se asignará ese valor en lugar del valor por defecto.
* ***<categoria>:*** es un String que nos indica la categoría, el grupo o la temática del ejercicio. Es un campo obligatorio para las hojas de estilo Wizard y WizardAdaptativo, en el caso de la hoja Clasica no es necesario y no se tendrá en cuenta.
* ***<Tipo\_de\_respuesta>:*** puede ser RespuestaUnica, RespuestaMultiple, TextoLibre u Ordenacion. Indica el tipo de respuesta del ejercicio.
* ***<correctas>:*** dependiendo del tipo de respuesta es posible escribir una lista de respuestas correctas, Strings separados por “,”. En el caso de la respuesta Ordenada se escribe la respuesta correcta ya ordenada en un solo string con “#” como separador. Por ejemplo: correctas {“1#2#3#4”}.
* ***<alternativas>:*** lista de respuestas alternativas, Strings separados por “,”.

### Validación

La validación del xText nos permite implementar restricciones más complejas que las posibles en el meta-modelo. En nuestro caso dejamos gran parte de las decisiones al usuario, por lo que no realizamos un gran control de errores. Pero mostramos warnings cuando el usuario escribe números negativos en alguna puntuación o en la penalización.

Añadimos la restricción de que las hojas de estilo Wizard y WizardAdaptativo deban incluir en los ejercicios la categoría a la que pertenecen, puesto que es un atributo necesario para agrupar los ejercicios y después mostrarlos.

Otra comprobación es que una opción correcta no sea también una opción alternativa, pero no se comprueba si se repiten opciones correctas o alternativas.

### Observaciones

No consideramos necesario generar un asistente de contenido, puesto que la gramática es sencilla y el asistente generado por defecto es suficiente.

La nota de la corrección puede ser menor que 0 si existe penalización y no se controlan los límites de la nota máxima y mínima.

El WizardAdaptativo no repite ejercicios cuando termina los ejercicios de una temática, pasa a la siguiente pantalla.

1. [Proyecto 1 – Ejercicios de auto-evaluación](https://moodle.uam.es/file.php/29113/proyectos/Proyecto_1_-_Ejercicios_de_auto-evaluacion.pdf) [↑](#footnote-ref-1)