

## Tareas de Servidor

### Daniel.

Para el sprint 3, mis tareas principales fueron las implementaciones del crud completo de reflexiones asociadas a las categorías principales del juego y además el alta de preguntas y respuestas que se utilizarán en el juego.

Para llevar a cabo estos requerimientos implementé el patrón MVC con el elemento de fachada (php/index.php), el cual es el encargado de recibir las distintas peticiones http y a partir de él se deriva a los distintos controladores y posteriormente a los distintos modelos.

Por otra parte destacar la funcionalidad de recibir peticiones AJAX con JSON en los bodys de las peticiones que también derivan siguiendo el anterior patrón y responde con un JSON creado a partir de consultas a la base de datos del juego.

La validación que se lleva a cabo tanto en el alta de preguntas como en el de reflexiones es el de los tamaños de la pregunta y reflexión.

El modelopreguntas.php y debido a la repartición de tareas, es usado tanto por mi compañera Laura como por mí mismo, usando cada uno nuestros propios métodos creados.

### Laura.

Modificación de preguntas y respuestas, borrado de preguntas y respuestas. Validación de servidor si las dos respuestas son iguales, la modificación no se registra. Usado el MVC, creada la fachada (php/hafacha/fachada.php) donde se llama a los controladores. Creado el controlador de modificación y además se usa el modelo de preguntas creado por Daniel (modelopreguntas.php). A parte se corrigen errores de sprint anteriores como añadir la ventana de confirmación de borrado tanto en preguntas como en subcategorías y visualizar las categorías de cada subcategoría. Se han creado vistas para la modificación y también para el borrado.

## David.

Creación de MVC para el inicio de sesión del administrador hacía la parte de CRUDs de la página. De este MVC:

- **Modelo:** Es dónde se realizan todas las consultas necesarias, y contiene todos los métodos necesarios:
  - login: Toma de parámetros el nombre y la contraseña. Y lo que hace es conectar con la BBDD y con una consulta preparada obtiene el nombre del administrador y el hash de la contraseña guardada en la BBDD. Después de comprobar que la consulta devuelva alguna fila, y que el hash de la contraseña de la BBDD y la contraseña pasada como parámetro coincidan, se crea una cookie con 600 segundos de vida, con el nombre del administrador, y con el perfil del administrador (GI)
  - altaAdmin: Para dar de alta al administrador en caso de que no haya uno existente. Toma de parámetros el nombre y la contraseña que tendrá el administrador. Conecta con la BBDD y mediante una consulta preparada inserta en la tabla 'Administrador' un nuevo registro con el nombre, y el hash de la contraseña pasada al método.
  - conectar: Inicia una conexión la BBDD, los datos de conexión se sacan de constantes definidas en un archivo de configuración.
  - checkAdmin: Comprueba mediante una consulta preparada si existe o no un administrador en la base de datos. Devuelve un booleano.
- **Vista:** Las vistas (indexalta.php, indexlogin.php) se comunican con el controlador.
- **Controlador:** El controlador tiene los mismos métodos que el modelo. Cada uno de estos métodos llaman al método respectivo del modelo.

## Juanma.

### SPRINT2 SERVIDOR

He realizado los CRUDS de Categorías(creación y modificación del icono) y de Subcategorías(creación, modificación, borrado y listado). He aplicado la subida de archivos así como la modificación de los mismos. También en el sprint3 realicé el listado del ranking por php, así como la inserción de los

jugadores en el ranking. No he usado MVC y las validaciones no me dió tiempo a realizarlas.