**Ejercicio 1.**  
  
Creación de API sobre symfony 5.  
  
  
**Aclaraciones:**  
  
Inicialmente me propuse trabajar sobre php8 y symfony6 (aunque específicamente se pedía versión 5 en el ejercicio, entendí que sería incluso preferible) pero encontré bastantes problemas aún, especialmente con uno de los bundles elegidos para el desarrollo (FosRestBundle), por lo que decidí realizarlo en php7.4 y symfony5.  
  
Aunque no se requería BBDD, no me suponía mayor problema y he decido persistir sobre mysql5.7 y usar como servidor nginx. Está todo montado sobre docker, por lo que no debería ser difícil ejecutar el proyecto.  
  
Existen comandos make preparados para diferentes tareas, como:

* Acceder al contenedor pricipal > make ssh
* Construir y levantar los contenedores > make build
* Instalar el proyecto > make install (este construye los contenedores e instala dependencias composer)

No estoy muy familiarizado con el estándar jsonApi, y aunque he visto que API-platform lo soporta y está ya listo para ser utilizado, no conozco API-platform muy a fondo y aparte como ejercicio he preferido crear algo más “artesanal”.  
  
Para lo cual, he dicido por una parte usar FosRestBundle, que te permite automatizar la serialización de las peticiones y las respuestas (Entre otras funciones como versionado de API etc no utilizadas en el ejercicio).  
  
También he construido DTO’s (Data transfer objects) tanto para las peticiones como para las respuestas. Así tanto la respuesta como la petición pueden ser distintas de la entidad, y las validaciones se pueden aplicar sobre el DTO de la petición.  
Aparte, para convertir el objeto de la respuesta al formato json-api se usa un normalizador.  
  
Así mismo las excepciones son transformadas igualmente al formato json-api en un eventListener.

El api está expuesto en localhost/api/ y se dispone de documentación swagger en localhost/apidoc/

**Cosas que me hubiera gustado entregar pero dejo pendientes por falta de tiempo:**

Caché, haciendo uso de memcached o redis por ejemplo.  
Test unitarios y funcionales.

Front.