ENTREGA 1

REQUISITOS DE ENTREA

* Proyecto Symfony < NombreAlumnoProyectoSymfony>
* El proyecto debe tener
  + Archivo env : configuración con la base de datos con nombre <spotifyDB>
* Archivos docker-compose y dockerFile con la configuración del proyecto
* Se debe crear el modelo de clases de entidades doctrine propuesto en clase
* Un controlador por cada entidad de datos <NombreEntidadController>
* Cada controlador de creación de entidades en la base de datos
  + Ruta /entidad/new
  + La entidad debe ser creada a modo de prueba y con datos de prueba
* Modelo de clases en alguna herramienta
* El proyecto debe estar gestionado por un sistema de control de las versiones tipo github

NORMAS

* Cada entrega de proyecto es individual
* Las entregas son obligatorias (en la fecha establecida) para poder aprobar el proyecto
* El alumno es responsable del contenido de la entrega. El profesor verificará la idoneidad de cada entrega

LISTA DE CONTROL DE LA ENTREGA

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Descripción** | **Cumple (✔/✘)** |

|  |  |  |
| --- | --- | --- |
| **Entregada en fecha** | La entrega está subida al aula virtal en la fecha establecida |  |
| **Configuración del proyecto** | Archivo .env configurado con la base de datos llamada spotifyDB. |  |

|  |  |  |
| --- | --- | --- |
| **Configuración Docker** | Archivo docker-compose.yml configurado correctamente. |  |

|  |  |  |
| --- | --- | --- |
|  | Archivo Dockerfile funcional y bien configurado. |  |

|  |  |  |
| --- | --- | --- |
| **Modelo de clases Doctrine** | Modelo de clases de entidades diseñado según lo propuesto en clase y funcional. |  |

|  |  |  |
| --- | --- | --- |
| **Controladores** | Un controlador creado para cada entidad, siguiendo la nomenclatura <NombreEntidad>Controller. |  |

|  |  |  |
| --- | --- | --- |
| **Rutas** | Ruta /entidad/new implementada correctamente para cada entidad. |  |

|  |  |  |
| --- | --- | --- |
| **Datos de prueba** | Cada entidad tiene datos de prueba para validación. |  |

|  |  |  |
| --- | --- | --- |
| **Modelo de clases (herramienta)** | Modelo de clases creado en alguna herramienta visual (ej. UML). |  |

|  |  |  |
| --- | --- | --- |
| **Gestión de versiones (GitHub)** | Proyecto gestionado mediante GitHub, con un repositorio accesible. |  |

|  |  |  |
| --- | --- | --- |
| **Funcionamiento general** | El proyecto es funcional y se ejecuta correctamente según los requerimientos. |  |