|  |
| --- |
|  |
| Practica PHP TDW |
|  |

|  |
| --- |
| Xin Chen Xu – bn0059 |

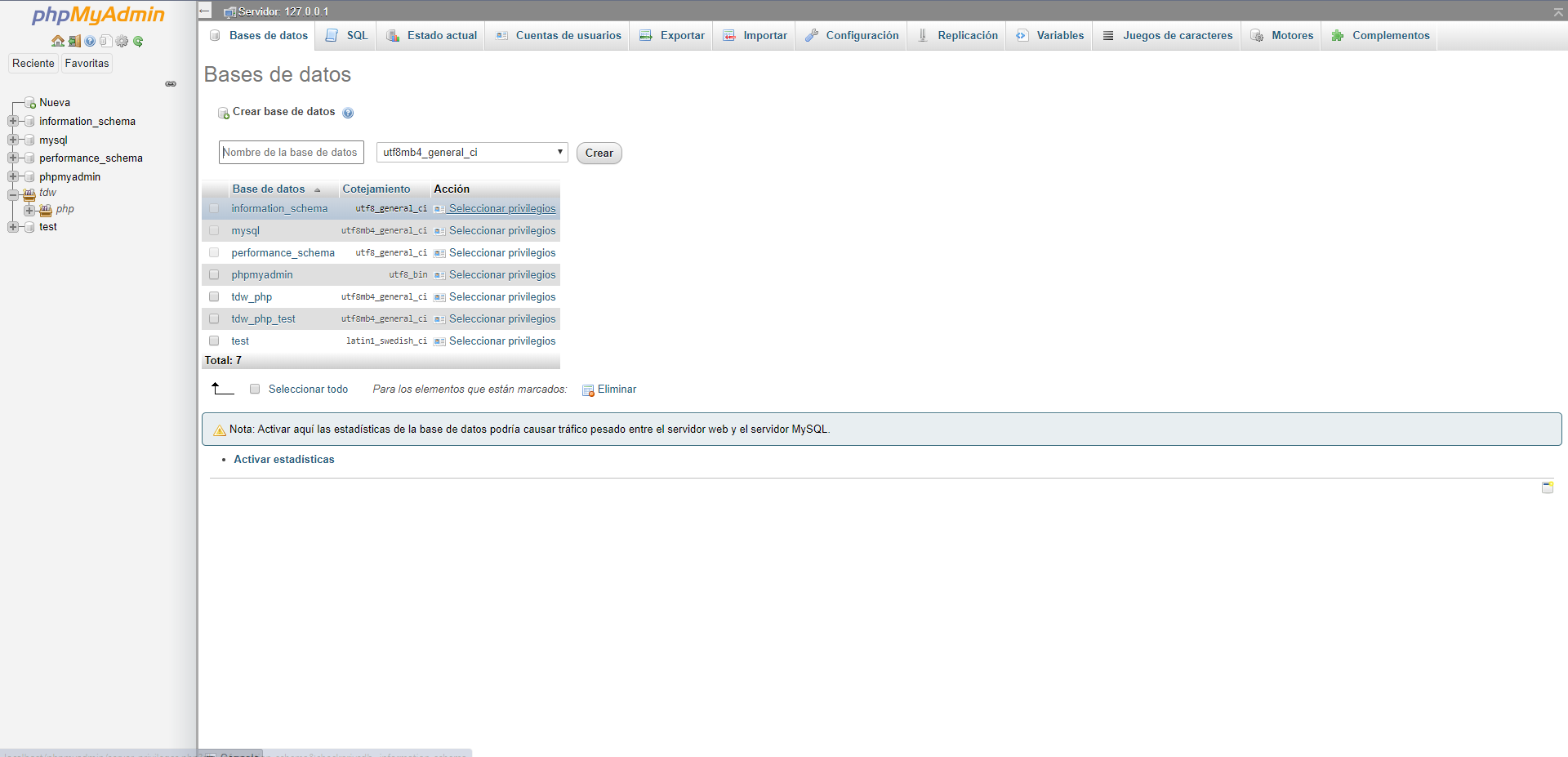
# Instrucciones para el despliegue de la practica

1.Instalar composer de forma que todos los proyectos puedan acceder uso a ella, se puede bajar de la web <https://getcomposer.org/download/> instalándolo de forma manual o usando un instalador que tienen.

2.Descomprimir la practica en X:\XAMPP\htdocs\xampp\Tema\_PHP preferiblemente para asegurar que no falle.

3.Abrir Xampp y tener activado apache y mysql.

4.En <http://localhost/phpmyadmin/> , se debe crear 2 bases de datos, una llamada tdw\_php y la otra tdw\_php\_test.



5.Tras la creación de las 2 bases de datos, se debe agregar una cuenta de usuario con el nombre adminUser y Password: \*adminUser\*, y marcando la casilla Privilegios globales seleccionar todo.

6.Abrir el proyecto con phpmystorm y ejecutar en la consola *composer install* 2 veces, para asegurar que se hayan creado las tablas y en este caso, insertado la cuenta de usuario AdminUser que se va a necesitar uso después.

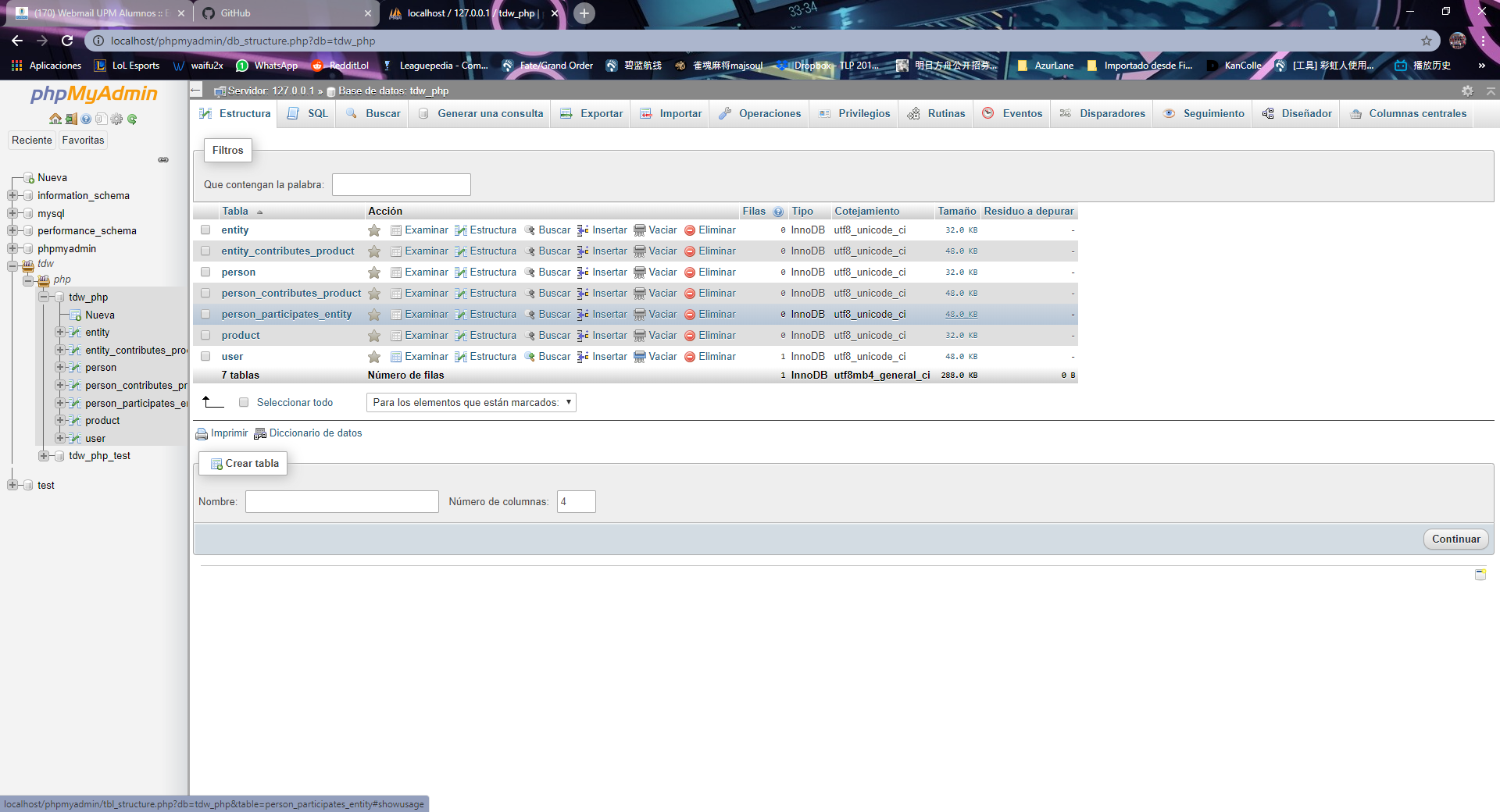
7.Ejecutar php -S 127.0.0.1:8000 -t public y abrir el hiperenlace que aparece en la consola para acceder a la api.

8.Loguearse con Usuario: adminUser y Password: \*adminUser\* para poder hacer uso de las funciones con candado

# Configuración del entorno de la práctica.

Como primer paso, se ha creado los archivos .env.local y phpunit.xml que con copias respectivas de .env y phpunit.xml.dist y se han hecho las modificaciones para que detecte la base de datos tdw\_php, tdw\_php\_test y el usuario adminUser que se ha creado previamente en phpmyadmin.

Tras ello, se ha usado el comando de *composer install* para instalar los componentes necesarios y tras eso, se ejecuta el comando por segunda vez para asegurar que se han metido los datos en la bbdd creada antes en phpmyadmin.



Probamos el conjunto de tests del proyecto y vemos que falla en entityController, entityRelationsController, personController y PersonRelationsController.

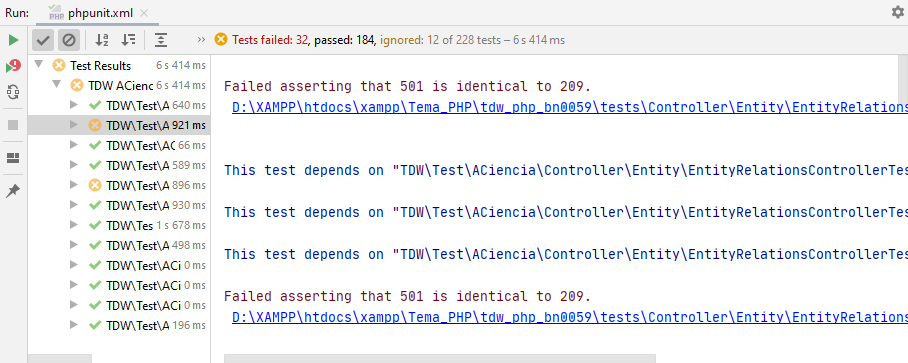
# EntityController

Se completa la parte de código de entityController, en concreto las funciones cget, get, getEntityname, delete, post y put tomando como ejemplo las de productController y se ejecuta una vez phpunit.xml para comprobar que esta correcto esa parte. Tras la comprobación se procede a hacer un push al repositorio github creado.



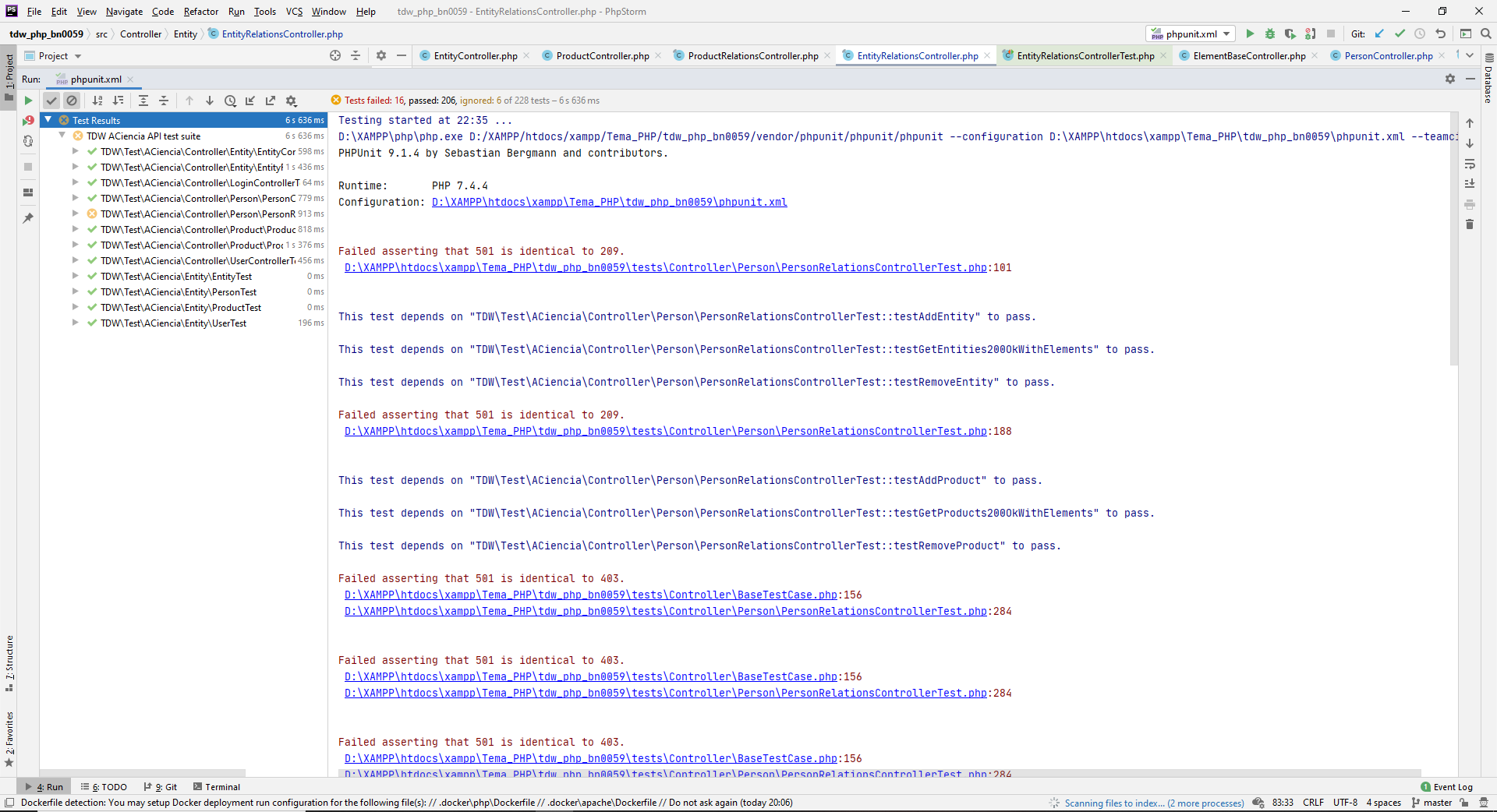
# PersonController

Se completa la parte de código de personController, en concreto las funciones cget, get, getEntityname, delete, post y put tomando como ejemplo las de productController y se ejecuta una vez phpunit.xml para comprobar que esta correcto esa parte. Tras la comprobación se procede a hacer un push al repositorio github creado.



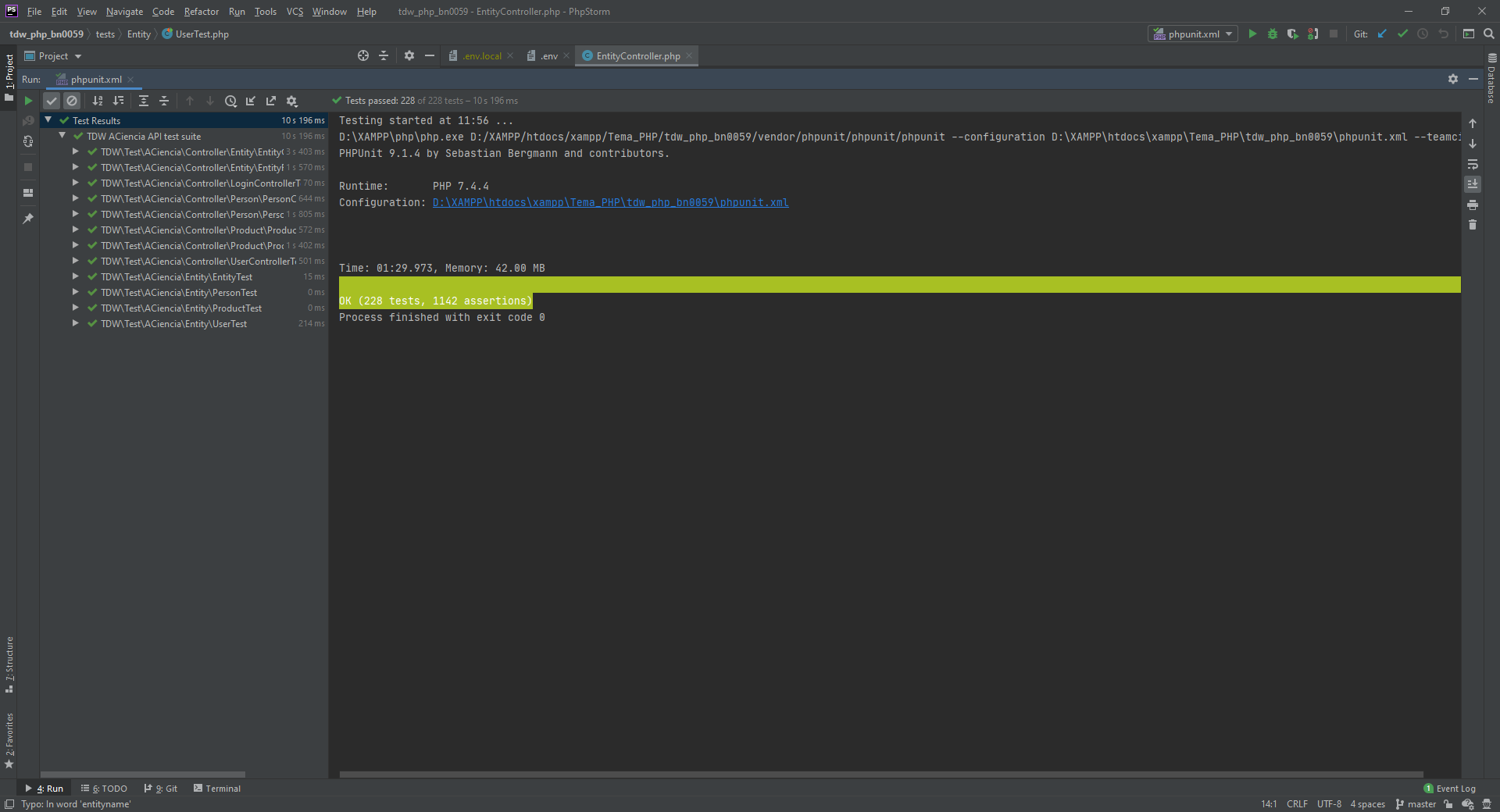
# EntityRelationsController

Se completa la parte de código de entityRelationsController, en concreto las funciones getPersons, operationPerson, getProducts y operationProduct tomando como ejemplo las de productRelationsController y se ejecuta una vez phpunit.xml para comprobar que esta correcto esa parte. Tras la comprobación se procede a hacer un push al repositorio github creado.



# PersonRelationsController

Se completa la parte de código de personRelationsController, en concreto las funciones getEntities, operationEntity, getProducts y operationProduct tomando como ejemplo las de productRelationsController y se ejecuta una vez phpunit.xml para comprobar que esta correcto esa parte y vemos que el proyecto ha pasado todos los tests.



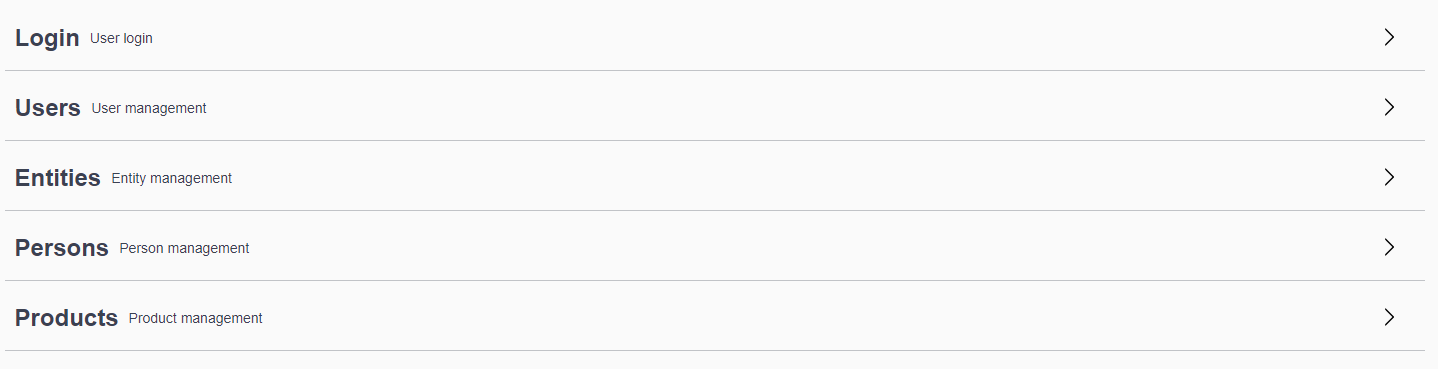
Tras ello, se procede a realizar un push al repositorio github y estaría completa la parte de los controladores.

# Modificación del api de la practica

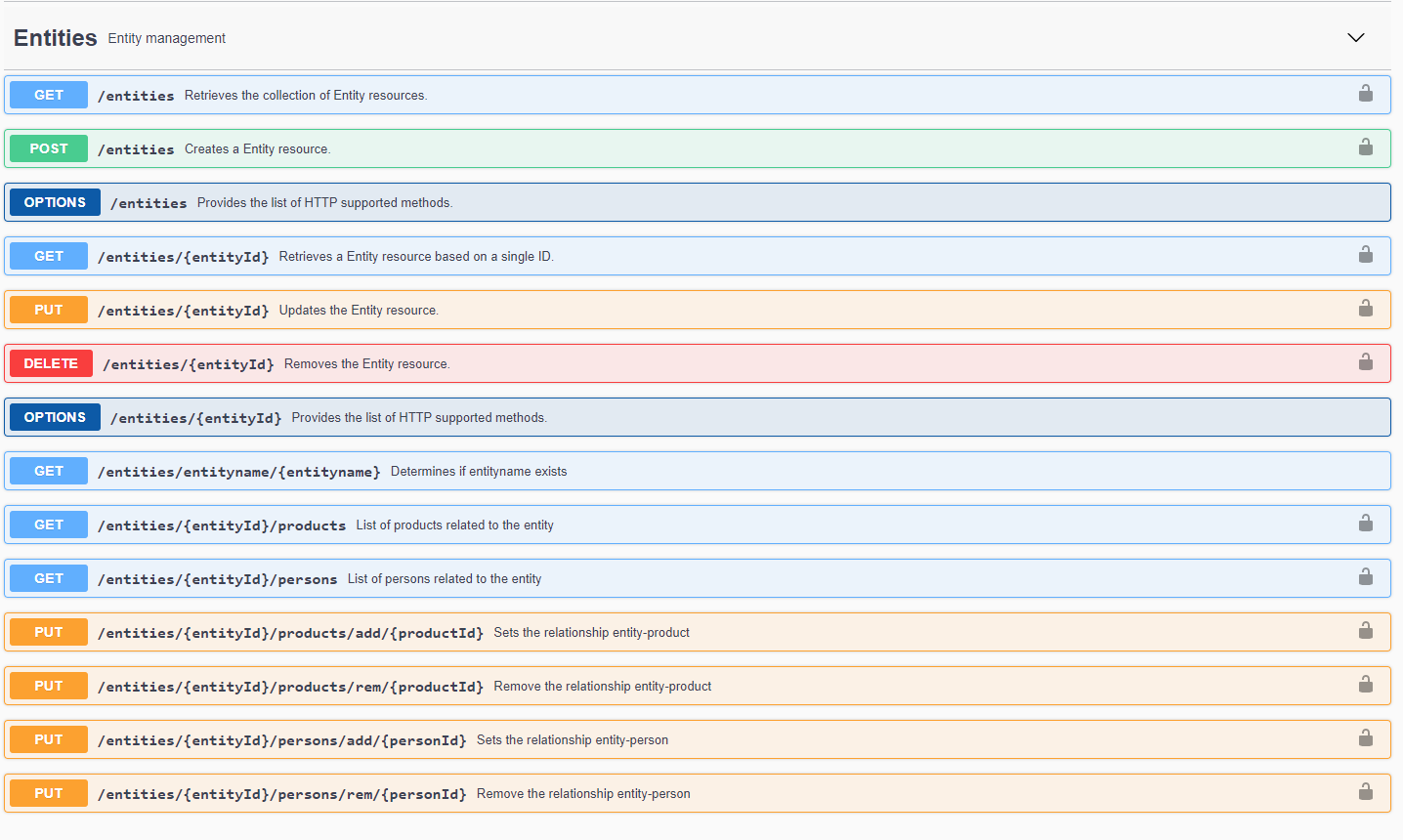
En el archivo openapi.yami, añadimos los tags correspondientes a Entities y Persons para que aparezcan al ejecutarlos y creamos todas las funciones necesarias para esas 2, en este caso también es necesario la creación de los parámetros entityname y productname que necesitan alguna de las funciones. Tras ello, se ejecuta la api y se procede a realizar la comprobación del correcto funcionamiento de todas las opciones que nos da la api.

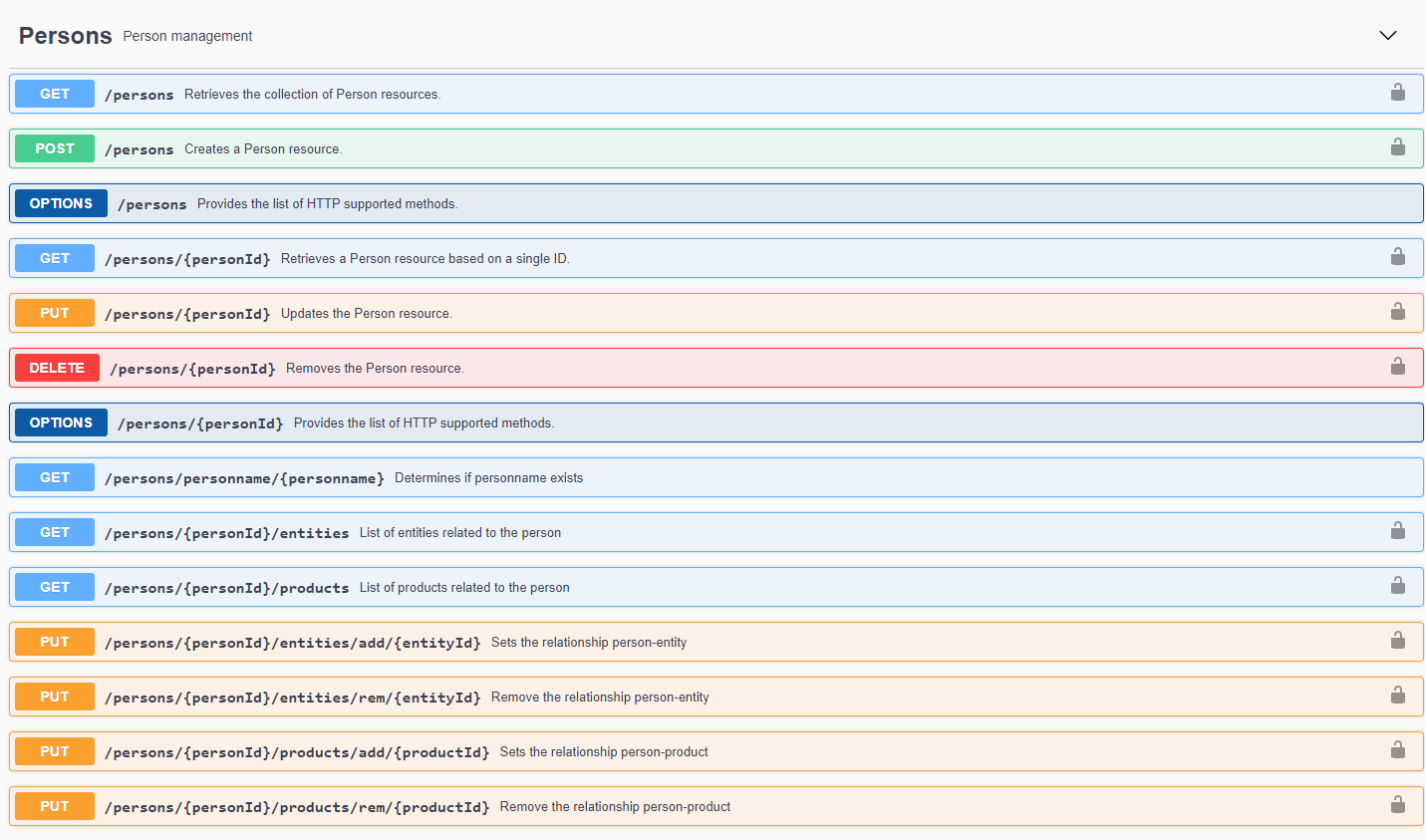
Se utiliza Usuario: adminUser y Password: \*adminUser\*

Api:



Entities:



Persons: 

Github: <https://github.com/XinChen97/tdw_php_bn0059>