



POLITÉCNICA



ETS I SISTEMAS  
INFORMÁTICOS

## MiW – Práctica 1 PHP

### Objetivo:

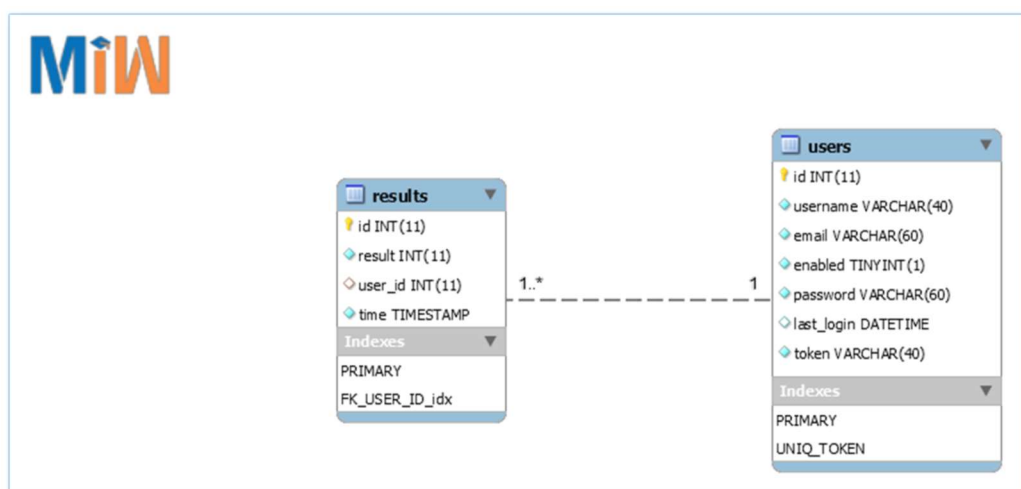
- Progresar en el conocimiento del lenguaje de scripting PHP y el ORM Doctrine y familiarizarse con el desarrollo de aplicaciones web completas integrando tecnologías del lado del cliente con el procesamiento en el lado del servidor.

### Enunciado:

Esta práctica pretende servir de ejemplo de integración de diferentes técnicas aprendidas en la asignatura “*Back-end con tecnologías de libre distribución (PHP)*”. En la misma se profundizará en el conocimiento del lenguaje de scripting PHP, se emplearán datos almacenados en un gestor de bases de datos y, adicionalmente, se utilizarán componentes desarrollados por terceros que permitirán simplificar las labores a desarrollar.

### Modelo de Datos:

El sencillo modelo de datos que se debe implementar (aunque se podrá adaptar libremente) será similar al siguiente:



## Se pide:

Empleando el ORM Doctrine<sup>1</sup>, definir las entidades correspondientes e implementar un conjunto de scripts (en el directorio `./src` del proyecto) que desde línea de comandos –y opcionalmente también vía web– permitan añadir, mostrar, actualizar y eliminar los diferentes usuarios y resultados. Los diferentes scripts mostrarán el resultado por pantalla y, en caso de recibir el parámetro `--json`, generarán la respuesta en formato JSON<sup>2</sup>.

Las operaciones añadir, actualizar y eliminar se aplicarán sobre un único elemento. Por su parte también se implementarán operaciones que permitan mostrar un único elemento y todos los elementos.

De igual forma, se deberán implementar las operaciones anteriores vía web (ver *Notas*).

## Opcional:

- Desarrollar scripts que permitan añadir/eliminar/modificar múltiples datos para las diferentes entidades
- Empleando el framework `phpunit`<sup>3</sup>, desarrollar pruebas unitarias y de integración del conjunto de entidades generadas (1 punto)

## Notas:

- Se podrán emplear cuantos componentes adicionales se estimen oportunos. Para el desarrollo vía web se sugiere emplear, por ejemplo, el componente `symfony/routing`<sup>4</sup> que simplifica el desarrollo del controlador frontal.

## Notas sobre la instalación:

La versión mínima necesaria del intérprete de PHP es la 5.5.0, si bien es recomendable actualizar a la última versión estable disponible (PHP 7.3.\* actualmente). Para conocer la versión disponible en el sistema:

```
1D:\xampp\htdocs> php --version
2PHP 7.3.11 (cli) (built: Oct 22 2019 11:20:18) ( ZTS MSVC15 (Visual C++ 2017) x64 )
3Copyright (c) 1997-2018 The PHP Group
4Zend Engine v3.3.11, Copyright (c) 1998-2018 Zend Technologies
5    with Xdebug v2.8.0, Copyright (c) 2002-2019, by Derick Rethans
6
```

---

<sup>1</sup> <http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/>

<sup>2</sup> <http://www.json.org/>

<sup>3</sup> <https://phpunit.de/>

<sup>4</sup> <http://symfony.com/doc/current/book/routing.html>

Una vez comprobada la versión del intérprete de PHP se procederá a la generación del esquema de base de datos (`miw_results`) y de un usuario y sus correspondientes permisos de acceso en el gestor de bases de datos. Desde cualquier cliente MySQL/MariaDB (por ejemplo, `phpMyAdmin` o `MySQL Workbench`) es posible realizar este paso (desde la opción de gestión de usuarios). En relación a este aspecto, el proyecto emplea el componente PHP `dotenv`<sup>i</sup>, por lo que los parámetros deberán ser establecidos en el fichero de configuración de variables de entorno (`/.env`). Por ejemplo:

```
1#
2# Configuración SGBD
3#
4DATABASE_HOST      = '127.0.0.1'
5DATABASE_NAME      = 'miw_results'
6DATABASE_PORT      = 3306
7DATABASE_USER      = 'miw'
8DATABASE_PASSWD    = '*miw*'
9DATABASE_DRIVER     = 'pdo_mysql'
10DATABASE_CHARSET   = 'UTF8'
11
12# Admin User data fixtures
13ADMIN_USER_NAME    = 'admin-miw'                # CAMBIAR!!!
14ADMIN_USER_EMAIL   = '${ADMIN_USER_NAME}@example.com'
15ADMIN_USER_PASSWD  = '*${ADMIN_USER_NAME}*'     # CAMBIAR!!!
16
17# Configuración Doctrine
18ENTITY_DIR         = './src/Entity/'
19DEBUG              = 0      # muestra consulta SQL por la salida estándar
20
```

A continuación empleando el gestor de dependencias `composer` se procederá a instalar y configurar los componentes requeridos por la aplicación. Este comando se ejecutará desde el directorio raíz del proyecto:

```
1D:\xampp\htdocs\MiW19\ResultsDoctrine> composer install
2Loading composer repositories with package information
3Installing dependencies...
4[...]
```

Con la ejecución del comando anterior se generará en el proyecto la carpeta `vendor`, que es donde residen los componentes desarrollados por terceros. Una vez hecho esto... ¡ya se puede empezar a desarrollar nuevas opciones!

---

<sup>i</sup> <https://github.com/vlucas/phpdotenv>