

EJERCICIO FRAMEWORK

Se poseen las siguientes tablas:

USUARIOS	FAVORITOS	USUARIOSPAGOS	PAGOS
codigousuario usuario clave edad	codigousuario codigousuariofavorito	codigopago codigousuario	codigopago importe fecha

Aclaración: la tabla "favoritos" une usuarios con otros usuarios a través de los campos codigousuario y codigousuariofavorito. Un ejemplo sería que si el usuario con código 1 tiene de favoritos a los usuarios de código 5,6 y 8 en la tabla figurara:

codigousuario	codigousuariofavorito
1	5
1	6
1	8

Utilizando algún framework:

cree las clases que le parezca conveniente tener, en relación a las tablas expuestas, para poder administrar (altas, bajas y modificaciones) los usuarios, sus favoritos, y sus pagos, teniendo en cuenta lo siguiente:

- Los usuarios no pueden tener un nombre de usuario vacío (campo "usuario" de la tabla Usuarios), y la edad debe ser mayor a 18.
- No deben poder crearse registros muertos (codigousuario inexistente en la tabla Usuarios) en la tabla Favoritos, teniendo en cuenta el control de sus dos campos.
- Mismo punto que para la tabla UsuariosPagos
- El importe de los Pagos debe ser mayor a cero, y la fecha no puede ser anterior al día de hoy.

Las clases deben contar con **todas las validaciones mencionadas**, de manera que un programador que no tenga acceso a la documentación igualmente pueda instanciar y utilizar los objetos de estas clases correctamente

Se valorará si se implementa interfaz de usuario.

Opcionalmente (siendo un plus que sumará puntos a la hora de la evaluación) se podrá incluir un archivo Dockerfile (Docker) que permita realizar el build de una imagen de container que contenga:

- Intérprete php.
- Dependencias (en caso de haberlas)
- Web Server (Apache o Nginx)
- Código del ejercicio.

El container deberá poder buidarse con el comando: `docker build -f Dockerfile -t framework .`

De igual modo se valorará la inclusión de un archivo docker-compose.yml (Docker Compose File) que permita orquestar dos servicios (containers): el container mencionado y otro con una Base de Datos necesaria al ejercicio (MySQL):

La orquestación deberá poder instanciarse mediante el comando: `docker-compose -f docker-compose.yml up`

El Servicio que incluye el Web Server deberá ser accesible mediante el puerto 80 del equipo anfitrión (`http://localhost`)