

GUIÓN DE LA ACTIVIDAD DE EVALUACIÓN 2 (AEV2):

Título

AEV2 - Modelo Vista Controlador con Doctrine

Descripción

Esta actividad evaluable consiste en crear una aplicación que cumpla los criterios vistos en los temas 1, 2, 3 y 4.

Debemos crear un proyecto que cumpla con el patrón MVC, que tenga un sistema de enrutamiento mediante el autoloader de composer y los namespaces. Usaremos TWIG para implementar las plantillas visuales. Y

Programa

1. Todo el bloque 1 (Tema 1 y 2) y bloque 2 (Tema 3 y 4)
2. La actividad tiene una temporalización de unas 16 horas aproximadamente.

Recursos generales

Presentaciones y videos de los temas 1, 2, 3 y 4:

- Introducción a MVC
- Introducción a TWIG
- Introducción a Doctrine ORM
- Doctrine Asociaciones
- CRUD con Doctrine

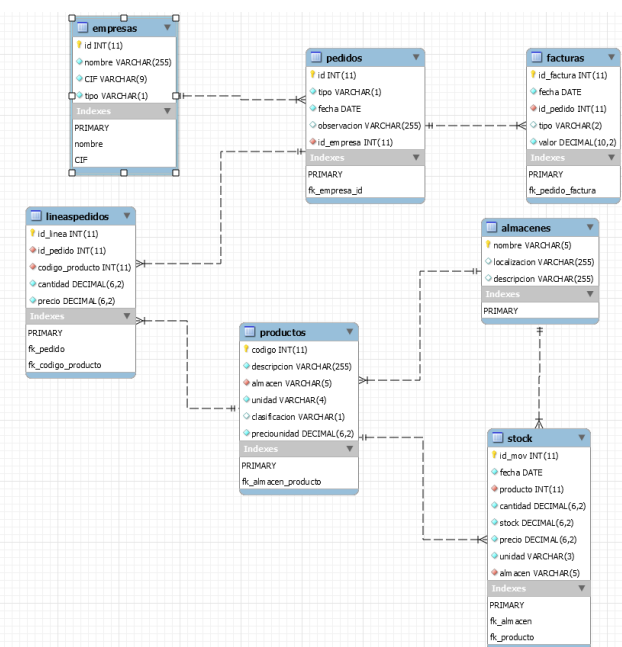
Material de apoyo:

- Todo el entregado en los temas.

Actividades

Para ello se pide:

1. Crear tres niveles de herencia, “abuelo, padre e hijo”, utilizando para el layout el formulario HTML entregado.
2. Usar el script adjunto para importar la BB.DD. corporación y modelar todas sus tablas con las asociaciones que contienen. Se tendrán en cuenta que todas las asociaciones deben ser bidireccionales, salvo la asociación entre pedidos y facturas que será unidireccional.
3. Crea un proyecto con la estructura típica del MVC que se ha visto hasta la fecha con Doctrine.
4. Instala Composer (el gestor de dependencias que vamos a usar todo el curso) e implantaremos las especificaciones de autoloader [PSR-4](#).
5. La configuración deberá guardarse en un archivo .env



6. Es importante que durante el desarrollo del proyecto se use un repositorio GIT, en este caso usaremos GitHub donde debéis crear un repositorio para este proyecto, privado y compartirlo conmigo para que yo pueda acceder, invitarme como fdiaz-alonso@florida-uni.es
7. Existirá un botón o link que nos permita volver a la raíz del proyecto "/" en todas las plantillas en la cabecera.
8. Existirá un botón o link que nos permita volver a la ruta anterior de donde hemos venido en la cabecera y en el footer.
9. La aplicación solo responderá a las siguientes rutas:
 - 9.1. **/o raíz →** Debe mostrar el template entregado renderizado con TWIG pero tal y como se vería si usáramos únicamente en HTML.
 - 9.2. **/products →** Cargará la lista de todos los productos. Modifica la plantilla TWIG usada en la ruta raíz y extiende de ella de una forma adecuada. La lista de los productos estará contenida en una tabla. No debe mostrarse en la tabla ninguna primary key y en las foreign key deben mostrarse en lugar del código el nombre o descripción de la fila que corresponda.
 - 9.3. **/wharehouses →** Cargará la lista de todos los almacenes en una tabla. Ha de reutilizarse la plantilla usada para el renderizado de productos.
 - 9.4. **/bunisses →** Cargará la lista de todas las empresas en una tabla. Ha de reutilizarse la plantilla usada tanto en productos como en almacenes. En la columna de tipo debe indicarse si es proveedor o cliente teniendo en cuenta el comentario en la BB.DD. para esa columna. Además, en esta plantilla tendremos la posibilidad de actualizar cualquier empresa mediante un link o botón en la tabla que nos lleve a la ruta **/update**. También podremos insertar una nueva empresa mediante un botón o link en la cabecera de la plantilla que nos debe llevar a la ruta **/insert**. También tendremos en cada línea de la tabla un botón o link de la plantilla para poder borrar una empresa mediante la ruta **/delete**.
 - 9.5. **/orders →** Cargará la lista de todos los pedidos en una tabla. Se ha de crear una nueva plantilla TWIG para este renderizado. No debe mostrarse el código de las foreign key, debe mostrarse el nombre o descripción de la fila que corresponda. La id de cada pedido nos servirá para mostrar el detalle mediante la ruta **/orderDetail**. Además, tendremos un botón o link que lleve a ver las facturas asociadas al pedido mediante la ruta **/invoice**.
 - 9.6. **/orderDetail →** Cargará todas las líneas de un pedido mediante el parámetro recibido por la ruta de la id del pedido. Se ha de mostrar los datos en una tabla. Hemos de reutilizar la plantilla usada en pedidos. Mostraremos de nuevo los datos del pedido en una tabla superior y debajo de esta mostraremos una nueva tabla con los datos de las líneas de pedidos. No debe mostrarse el código de la foreign key, de producto, debe mostrarse el nombre o descripción de la fila que corresponda.
 - 9.7. **/invoices →** Cargará una tabla con todas las facturas. Se reutilizará la plantilla usada en pedidos. Además, tendremos un link en la id de la factura para mostrarnos los datos de la misma en la ruta **/invoice**.
 - 9.8. **/invoice →** Cargará una tabla con el dato del pedido asociado mediante el parámetro recibido por las variables de sesión y cookies. Debajo una nueva tabla con los datos de la factura. Ha de reutilizarse la plantilla usada en pedidos.
 - 9.9. **/update →** Habremos recibido por las variables de sesión cual es la id de la empresa que se quiere actualizar. Podrá actualizarse de la empresa cualquier dato, salvo la id. Usar el HTML entregado como formulario para poder actualizar los datos, puede modificarse el formulario si es necesario.
 - 9.10. **/insert →** Insertaremos una nueva empresa usando el HTML entregado como formulario para poder insertar los datos, puede modificarse el formulario si es necesario.
 - 9.11. **/delete →** **Ojo es posible que tengas que modificar la estructura de la BB.DD. para poder borrar un dato si está asociado.** Usaremos una plantilla nueva que nos muestra un mensaje de confirmación para verificar que de verdad queremos borrarlo o no. Tras ello, nos debe redirigir a la ruta de empresas de nuevo.

- 9.12. **Cualquier otra ruta** deberá implementarse una plantilla nueva para recoger el error de ruta. Indicando que la ruta no está definida y poniendo un botón o link para ir a la raíz del proyecto
10. **Formato TWIG** → Hay que formatear:
 - 10.1. Las fechas al formato dd/mes con letra/aaaa usando **los filtros TWIG**.
 - 10.2. Los textos deben estar todos en minúsculas, salvo la primera letra de títulos y nombres que estará en mayúsculas, para hacer esto debemos **usar filtros TWIG, no lógica de PHP**.
 - 10.3. Todos los valores económicos de dinero deberán estar filtrados con **el símbolo de € (no usar filtros TWIG) y deberán mostrar siempre dos decimales mediante filtros TWIG**.
11. Realizar memoria con el formato que está en Florida Oberta, rellenando todos los puntos que indica.
12. Realizar video explicando cómo funciona la aplicación desarrollada de entre 10 y 15 minutos. El video mostrará como se ejecuta la aplicación, y todas sus funcionalidades arrancándola desde 0. Debéis salir con la cámara activa en alguna esquina del video durante todo el video. Al igual que la explicación debéis darla vosotros, intentando evitar leer un documento. Recordar preparaos el video.
13. Entregar Auto rúbrica con vuestra propia evaluación, usando la rúbrica entregada.

Recursos adicionales

- Usar script creación BB.DD. **corporacion.sql** en formato SQL que se ha entregado junto con los recursos de la actividad.
- Hoja de rubrica para rellenar la auto rúbrica y conocer los diferentes niveles de evaluación de la actividad.

Entregar:

- Fichero ZIP con la estructura completa del proyecto, incluyendo todos sus archivos. **No incluir la carpeta vendor.**
- Incluir la ruta al repositorio de Git que hayáis usado para desarrollar el proyecto, asegurarse que el repositorio es privado y esta compartido conmigo.
- Incluir ruta con el link de One Drive donde habéis subido el video y aseguraros que esta compartido conmigo y puedo acceder.
- Incluir archivo en formato Word o similar incluyendo la portada.

Rúbrica:

Existen cuatro niveles de corrección para esta actividad:

- **Mínimos (hasta 5 puntos)**
 - La aplicación debe conectar con la BB.DD.
 - Deberán estar implementados los **puntos del 1 al 8** de la sección de actividades más los puntos:
 - **Punto 9.1** → Que la ruta raíz, "/", se muestre al iniciar el proyecto desde la carpeta public, ejecutando los requisitos.
 - **Punto 9.12** → Deberá estar implementado que si se introduce una ruta no definida en la aplicación debe mostrar un mensaje como indica el punto 6.
 - **Punto 10 y subpuntos** → Usar el formato predefinido para TWIG.
 - **Punto 11** → Realizar memoria según documentación subida en Florida Oberta.
 - **Punto 12** → Realizar video explicativo de la actividad.
 - **Punto 13** → Rellenar la Auto rúbrica
- **Suficiencia (hasta 6 puntos)**
 - Cumplir todos los requisitos del punto mínimos.
 - Añadir los puntos **9.2** → /products, **9.3** → /warehouses y **9.4** → /bunisses.
- **Notoriedad (hasta 8 puntos)**
 - Cumplir todos los requisitos del punto suficiencia.
 - Añadir los puntos **9.5** → /orders, **9.9** → /update y **9.10** → /insert.
- **Excelencia (hasta 10 puntos)**
 - Cumplir todos los requisitos del punto notoriedad.
 - Añadir los puntos **9.6** → /ordersDetail, **9.7** → /invoices, **9.8** → /invoice y **9.11** → /delete.

Tabla de corrección de rúbrica y Auto rúbrica

Nivel de Rúbrica	Superado	No superado	Observaciones
Mínimos			
Suficiencia			
Notoriedad			
Excelencia			