

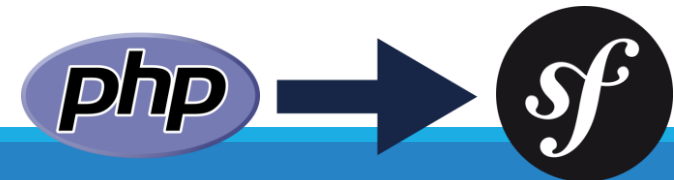
UD9. Aplicación de pedidos en Symfony

DESARROLLO WEB EN ENTORNO SERVIDOR

Aplicación con Symfony

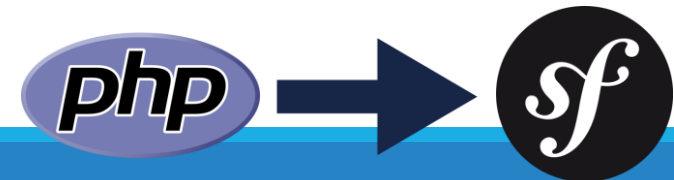
Vemos la aplicación de pedidos rediseñada utilizando Symfony:

- Misma apariencia y funcionalidad.
- Con un buen diseño y conociendo el framework → desarrollo más rápido.



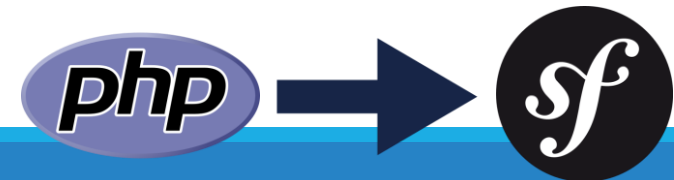
Partes que **no** cambian

1. La base de datos.
2. El mapa de pantallas.
3. La variable de sesión para el carrito.



Partes que **sí** cambian

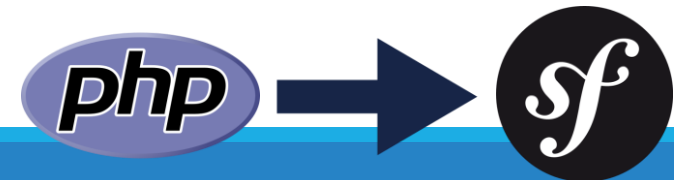
1. La salida → plantillas Twig.
2. La base de datos → con Doctrine.
3. Los archivos originales → rutas asociadas a controladores.



Rutas de la aplicación

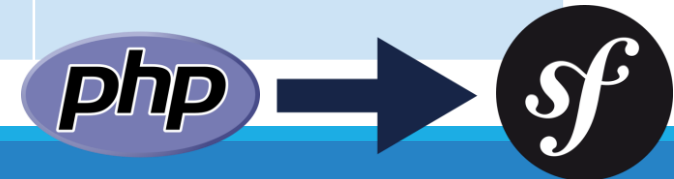
Hay que transformar los archivos de aplicación de la UD4 en controladores.

- Si la salida era un HTML → plantilla Twig.
- Si la salida era una redirección → tenemos que ajustar la ruta.



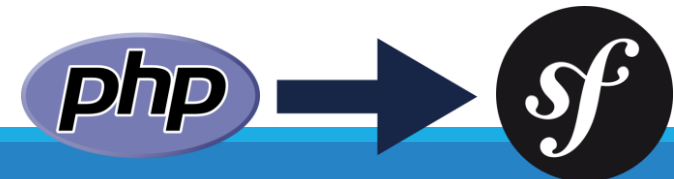
Rutas de la aplicación

Ruta	Descripción	Parámetros	Plantilla/redirección
/login name = 'ctrl_login'	Formulario de login	\$_POST['user'] \$_POST['password']	login.html.twig
/logout name = 'ctrl_logout'	Cierra la sesión		/login
/categorias name = 'categorias'	Crea una lista con las categorías, que enlazan a /productos/{id}		categorias.html.twig
/productos/{id} name = 'productos'	Muestra los productos de esa categoría. Permite añadir productos al carrito.	El código de la categoría	productos.html.twig
/carrito name = 'carrito'	Muestra el carrito. Permite quitar un producto.		carrito.html.twig



Rutas de la aplicación

Ruta	Descripción	Parámetros	Plantilla/redirección
/anadir name = 'anadir'	Añade elementos al carrito	\$_POST['cod'] \$_POST['unidades']	/carrito
/eliminar name = 'eliminar'	Quita elementos del carrito	\$_POST['cod'] \$_POST['unidades']	/carrito
/realizarPedido name = 'realizarPedido'	Inserta el pedido en la base de datos, envía el mail de confirmación y muestra un mensaje de éxito o error.		order.html.twig mail.html.twig

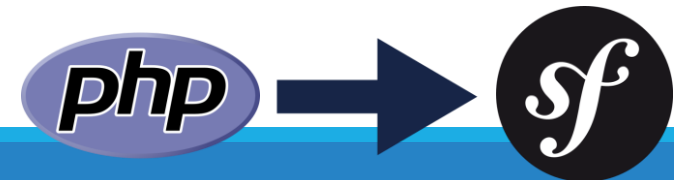


Plantillas

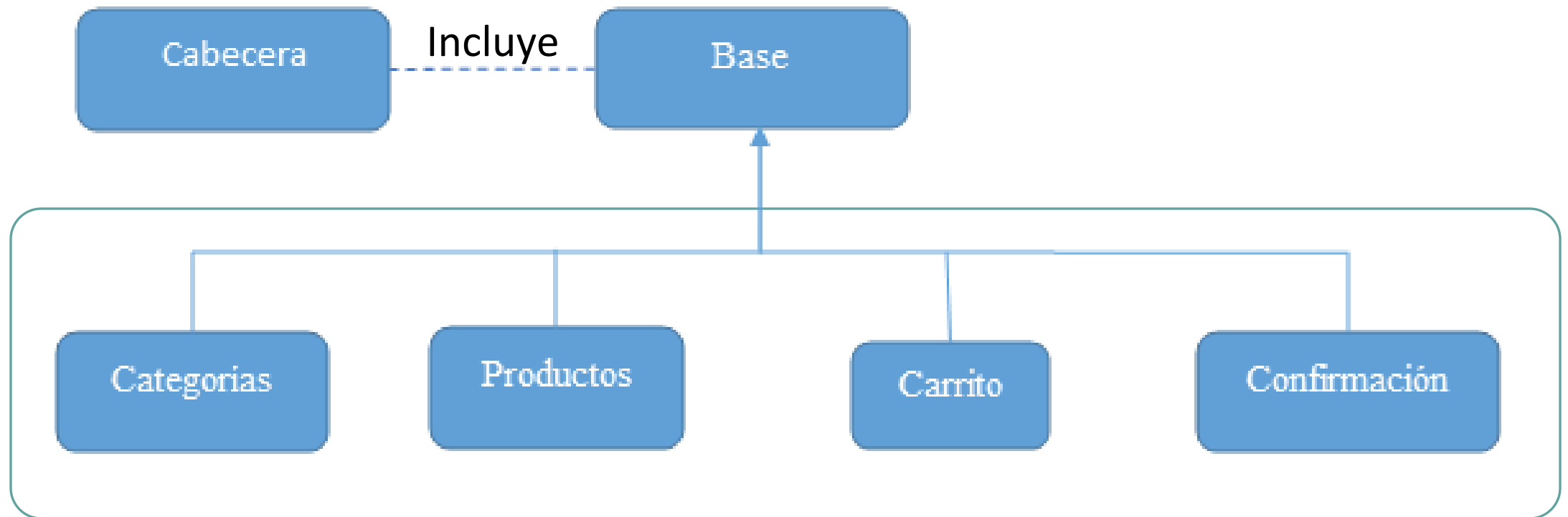
Tras el acceso, todas las páginas de la aplicación tienen una estructura común:

- Una cabecera en la parte superior.
- Una sección principal cuyo contenido varía.

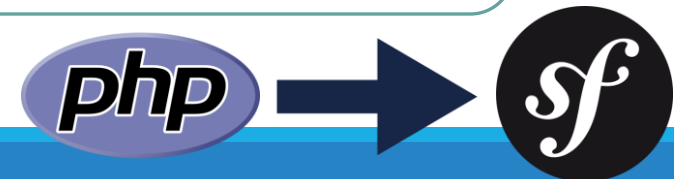
Para representarla, se utilizará una jerarquía de plantillas.



Plantillas



Heredan



Entidades

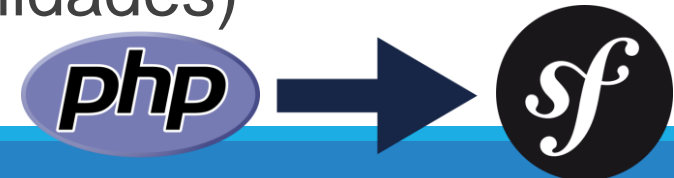
Restaurantes(CodRes, Correo, Clave, País, CP, Ciudad, Dirección)

Pedidos(CodPed, Fecha, Enviado, *Restaurante*)

Productos(CodProd, Nombre, Descripción, Peso, Stock, *Categoría*)

Categorías(CodCat, Nombre, Descripción)

PedidosProductos(CodPedProd, *Pedido*, *Producto*, Unidades)



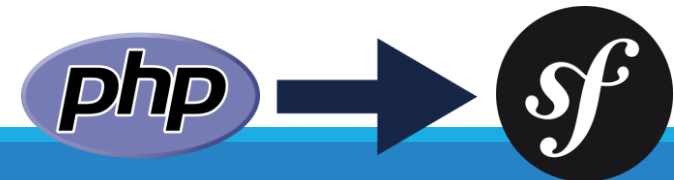
Entidades

Hay una entidad para cada tabla.

La entidad **Restaurante** es especial: representa los usuarios y se usa por el componente de seguridad de Symfony.

Asociaciones:

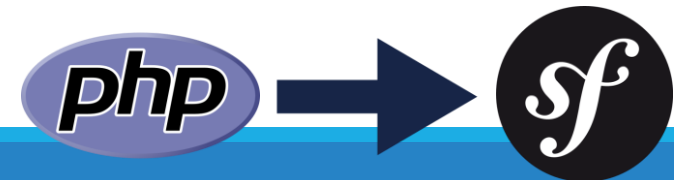
- Entre las entidades **Categoría** y **Producto** hay una asociación bidireccional.
- Entre las entidades **Pedido** y **Restaurante** hay una asociación unidireccional.
 - Cada pedido tiene una referencia al restaurante que lo realiza, pero no al revés.
- **PedidoProducto** tiene una asociación unidireccional a **Pedido** y **Producto**



Seguridad

Se usa el componente de seguridad de Symfony:

- Usuarios → entidad **Restaurante**.
- Roles → solo usaremos el por defecto, `ROLE_USER`.
- Formularios → indicados en el `security.yaml`.



Control de acceso

La clase `PedidosBase` contiene todos los controladores, excepto los de login y logout.

Estará protegida con el atributo PHP8:

```
#[IsGranted('ROLE_USER')]
class OrdersBase extends AbstractController
```

