APLICACIÓN CRUD CON LARAVEL

Descripción de la práctica:

Vas a realizar una aplicación CRUD en PHP con el framework Laravel:

1. Crearás una nueva base de datos llamada **pizzeria**
2. Crearás un nuevo proyecto laravel con composer, llamado **pizzaNombre**, siendo Nombre tu nombre de pila.
3. Utilizarás el framework Laravel.
4. La arquitectura será MVC: Definirás rutas, controlador, modelo y vistas con ayuda del framework.

La práctica consta de varios ejercicios, que irás resolviendo en orden.

Especificaciones y diseño:

Han abierto una nueva pizzería en el barrio, que atiende pedidos telefónicos.

Vas a desarrollar una aplicación web con PHP para los empleados, que permitirá:

1. Gestionar los datos de los clientes
2. Gestionar los pedidos de los clientes
3. Gestionar los datos de los tipos de pizza que ofrecen

Modelo de datos:

Hay tres entidades: Clientes (customers), pedidos(orders) y pizzas

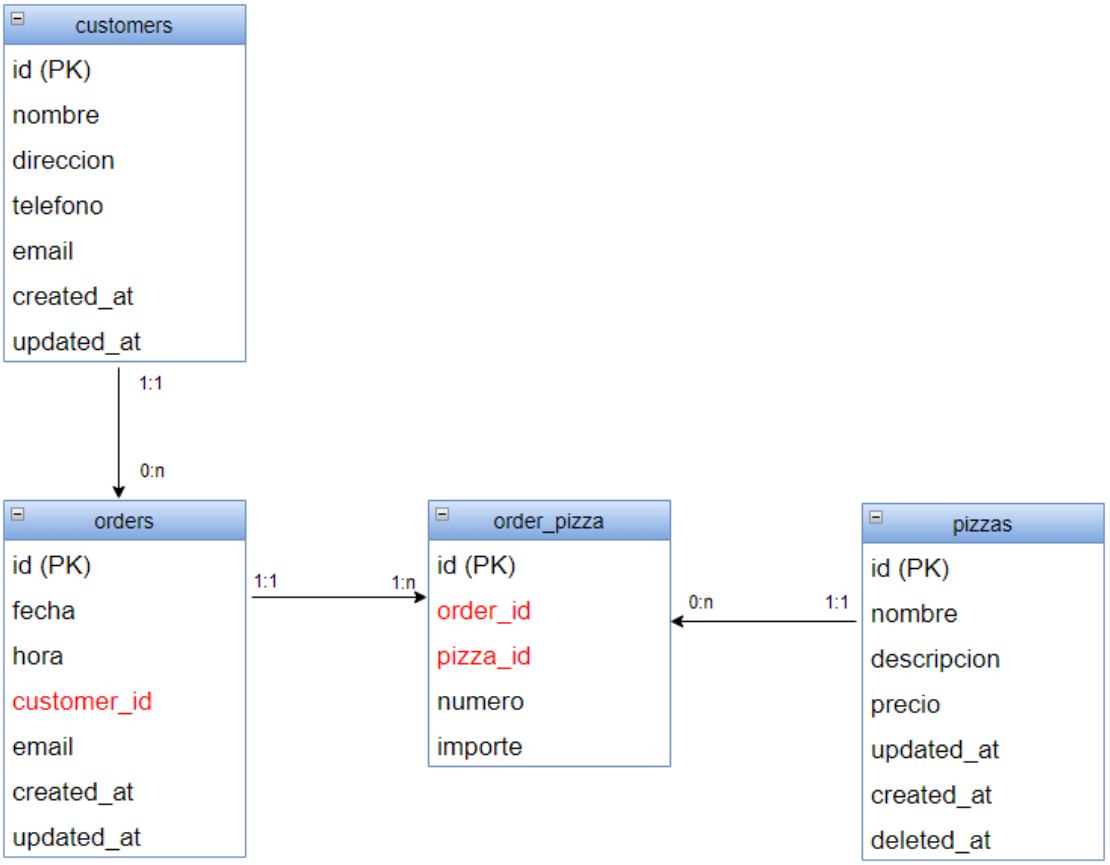
Hay dos relaciones:

- Un cliente puede tener varios pedidos (relación 1 a n)
- Un pedido puede incluir uno o más tipos de pizzas, y un tipo de pizza puede estar en varios pedidos (relación n a n)

El modelo está normalizado. En él pueden verse:

- Las claves primarias (PK)
- Las foreign keys (en rojo)
- La tabla que normaliza la relación n a n (order_pizza)
- Los nombres de las tablas y las claves primarias y foreign siguen la convención por defecto de Laravel. Se recomienda mantenerlos.

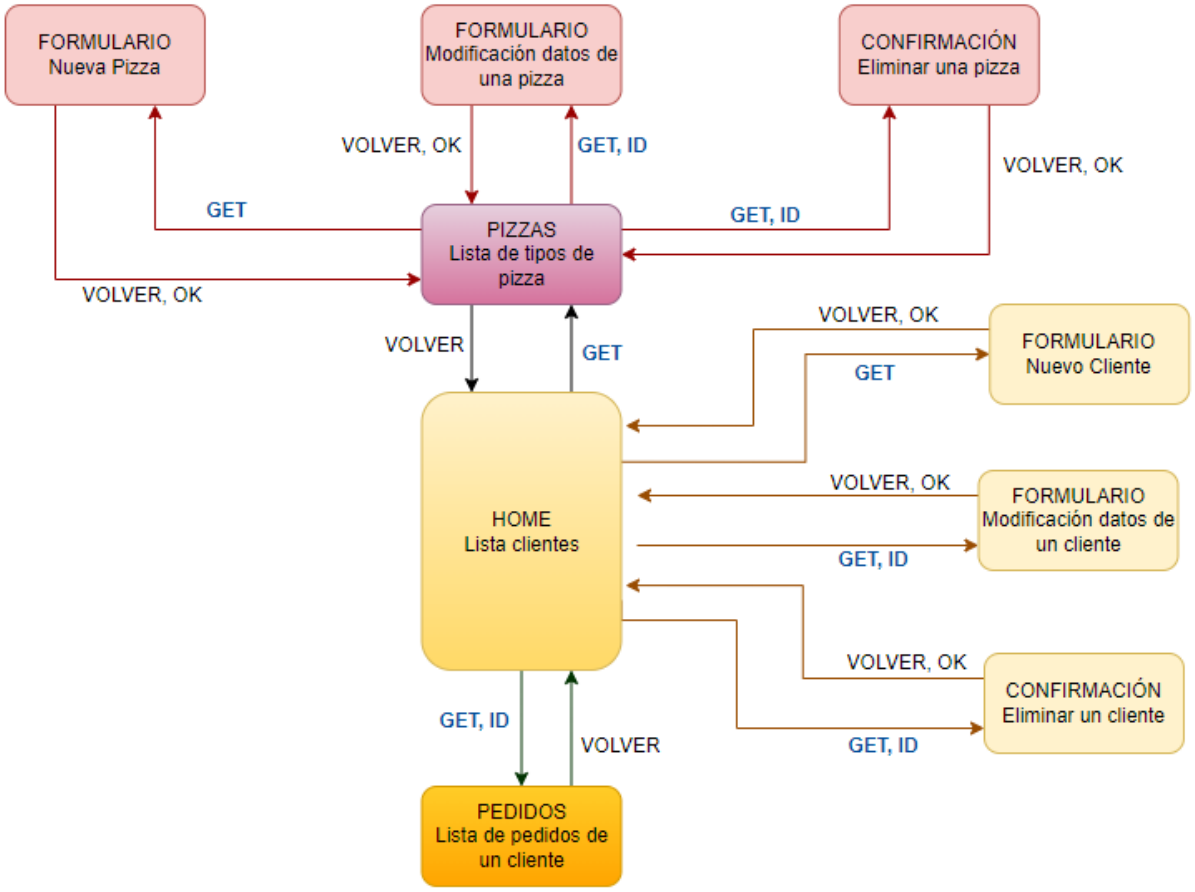
Gráfico del modelo de datos:



Notas:

- La tabla de pizzas utilizará borrado suave.
- Todas las tablas, a excepción de order_pizza, tendrán timestamps
- Todas las claves primarias son autoincrementadas

Gráfico de navegación entre páginas (vistas):



Notas:

- VOLVER: Se vuelve con GET a la página anterior. El botón “volver” estará disponible en todos los formularios de entrada de datos.
- Si un formulario produce un error, se retorna al formulario con GET, y se muestran los mensajes flash de los errores
- *Todas* las vistas se piden con GET.

EJERCICIO 1: Modelo

Vas a realizar las migraciones y los modelos necesarios para crear las tablas e implementar la aplicación. Todas las tablas se crearán con artisan.

Da de alta algunos datos para que no estén vacías cuando empieces a probar tu código.

EJERCICIO 2: Rutas para el CRUD de los clientes

Implementarás las siguientes rutas automáticas con **resource**, que se asociarán a los métodos de un controlador **Resource** de la forma:

Ruta	Method	ClienteController	Acción
`/clientes`	GET	index()	Muestra una página principal con todos los clientes y un menú con dos opciones: Clientes y Pizzas
`/clientes/{id}/edit`	GET	edit()	Muestra el formulario de modificación de datos de un cliente
`/clientes/create`	GET	create()	Muestra el formulario que permite introducir los datos de un nuevo cliente
`/clientes/{id}/delete`	GET	confirmar_borrado()	Muestra el formulario de confirmación de borrado de un cliente
`/clientes/{id}`	DELETE	destroy()	Borra el cliente y redirecciona a home
`/clientes({id}`	PUT	update()	Modifica los datos del cliente y redirecciona a home
`/clientes`	POST	store()	Crea un nuevo cliente y redirecciona a home

EJERCICIO 3: Controlador de la gestión de clientes

Crearás un controlador **ClienteController.php** para dar de alta las funciones que manipularán cada una de las rutas:

- Las rutas con GET retornarán una vista. Como ves, el CRUD de clientes necesita cuatro vistas.
- Las rutas POST, PUT y DELETE, realizarán la acción indicada y redireccionarán a /clientes.

Las funciones estarán inicialmente vacías, retornando únicamente un mensaje indicando lo que van a hacer. Prueba todas las rutas.
Nota: Puedes utilizar Postman o curl para probar las rutas DELETE, PUT y POST.

EJERCICIO 4: Vista “home” de la aplicación: /clientes

Harás una **plantilla** común a todas las vistas de la aplicación con:

- Un menú superior común a todas las páginas.
- El menú mostrará a la izquierda el logo/nombre de la pizzería, y en el centro dos opciones: “Clientes” y “Pizzas”.
- El contenido principal de la página se personalizará en cada vista específica

Harás la vista correspondiente a “Clientes”, que es la que se muestra con la ruta “`/clientes`”. Esta vista:

- Mostrará una lista con todos los clientes, ordenados alfabéticamente.
- Mostrarás 15 clientes por página. La paginación aparecerá en la parte superior, debajo del menú, y también al final de la página.
- Al principio de la página, mostrarás un botón con la opción de dar de alta un nuevo cliente.
- De cada cliente mostrarás nombre, dirección, teléfono e email.
- Junto a los datos de un cliente, aparecerán tres botones, uno para modificarlo, otro para eliminarlo, y otro para gestionar sus pedidos.

EJERCICIO 5: Gestión de clientes

5.1.- Alta de nuevo cliente

Realizarás la vista para el alta de un nuevo cliente, consistente en un formulario que solicita el nombre, la dirección, el teléfono y el email.

El submit enviará la información con el método POST para la validación de los datos y el alta del nuevo cliente.

Se validarán todos los campos (longitud máxima 255, no nulos, email válido):

- Si no son validados, retornarás al formulario, manteniendo la persistencia y mostrando los errores.
- Si son validados, darás de alta el nuevo cliente en la tabla customers, y volverás a /clientes.

5.2.- Modificación de cliente

Realizarás la vista para la modificación de los datos de un cliente, consistente en un formulario que muestra nombre, dirección, teléfono e email del cliente, y permite su modificación.

El submit enviará la información con el método PUT para la validación de los datos y su modificación en la base de datos.

Se validarán todos los campos (longitud máxima 255, no nulos, email válido):

- Si no son validados, retornarás al formulario, manteniendo la persistencia y mostrando los errores.
- Si son validados, modificarás los datos del cliente en la tabla customers, y volverás a /clientes.

5.3.- Borrado de un cliente

Realizarás la vista para confirmar la eliminación de un cliente. Esta vista mostrará:

- El nombre del cliente
- Un formulario que enviará la petición de borrar el cliente con DELETE, si se confirma el borrado.
- Una opción para cancelar, que retornará a /clientes

EJERCICIO 6: CRUD de los tipos de pizza

En este ejercicio, vas a realizar el CRUD de las pizzas ofertadas en la pizzería.

La opción de menú llamada “Pizza” llevará a una vista que mostrará:

- Mostrará una lista con todas las pizzas, ordenadas alfabéticamente por nombre.
- Mostrarás 10 pizzas por página. La paginación aparecerá en la parte superior, debajo del menú.
- Al principio, mostrarás un botón con la opción de dar de alta una nueva pizza
- De cada pizza, mostrarás nombre, descripción y precio
- Junto a los datos de una pizza, aparecerán dos botones, uno para modificarla, y otro para eliminarla.

Para este ejercicio realizarás un controlador llamado PizzaController, y diseñarás las rutas y las vistas para el CRUD. Utiliza rutas y controlador --resource.

No olvides validar los datos y redireccionar adecuadamente.

EJERCICIO 7: Consulta de los pedidos

El botón “Pedidos” que hay en la página principal asociado a cada cliente, navegará a la vista de pedidos del cliente.

Esta vista, mostrará:

- El nombre del cliente arriba del todo
- La lista de todos sus pedidos, ordenados de más reciente a menos reciente
- De cada pedido se indicará fecha y hora

EJERCICIO 8 (opcional): CRUD de los pedidos

¿ Te animas a realizar el diseño y CRUD de la gestión de los pedidos de un cliente ?. Ten cuidado, pues tienes una relación n a n.