

Examen de Laravel

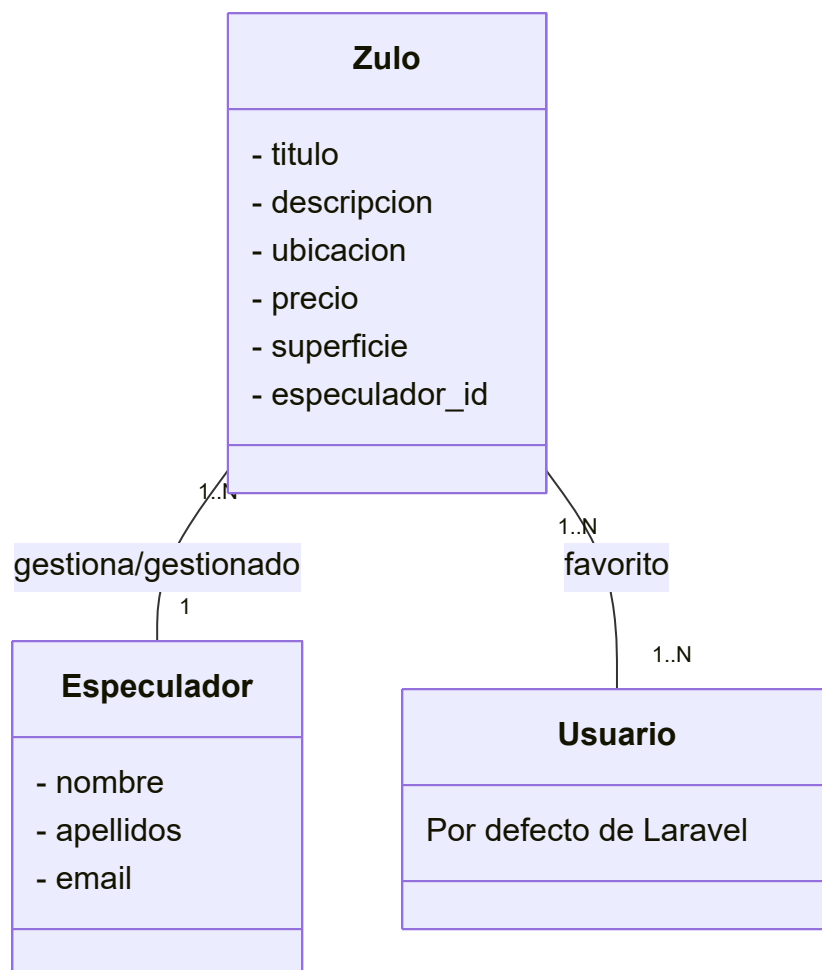
Fotozulo: alquiler de zulos, compra y venta

En la época de la precariedad en el acceso a la vivienda llega la solución a todos nuestros problemas:

Fotozulo. Se trata de una plataforma donde los usuarios pueden anunciar nuevas opciones de alquiler. Aquí todo vale: ¿tienes una despensa que no usas? ¡Métele un colchón hinchable y sácale 200€ al mes! ¿Un armario empotrado y apenas tienes ropa? Estás perdiendo dinero, amigo.

Modelo de datos

Un **especulador** puede dar de alta en la plataforma muchos **zulos**. Sin embargo, cada **zulo** sólo está gestionado por un **especulador**. Un **usuario** puede guardar en **favoritos** todos los **zulos** que quiera para que le llegue un aviso al email cuando se producen cambios en el precio. De la misma manera, un **zulo** puede estar en la lista de **favoritos** de varios **usuarios**.



Zulo

- **Título**, **Descripción** y **Ubicación** son cadenas de texto
- **Precio** es decimal
- **Superficie** es entero

Especulador

- **Nombre, Apellidos y Email** son cadenas de texto

1. Migraciones (2/10)

- Crea **modelo, migraciones y controladores**. Modifica las migraciones para representar el modelo de datos descrito el [Modelo de datos \(1.25/2\)](#).
- Adjunta un documento con los comandos ejecutados **(0.75/2)**.

Nota: no es necesario generar datos para la solución del examen, pero hazlo si crees que te ayuda a probar la aplicación

2. Relaciones (2/10)

Crea las siguientes relaciones:

- En **Especulador** :
 - **zulos()** : devuelve la lista de todos los zulos que gestiona
- En **Zulo** :
 - **especulador()** : quién gestiona el zulo
 - **interesados()** : qué usuarios están interesados (han guardado el zulo como favorito)
- En **User** :
 - **zulos()** : todos los zulos guardados como favorito

3. Rutas, Vistas y Controladores (4/10)

Mapea, usando **una sola línea**, todas las peticiones estándar de la convención RESTful al **controlador de Zulos (0.5/4)**.

Implementa lo siguiente:

1. Listado principal (**index**) **(0.75/4)**:
 - Muestra todos los zulos. En cada zulo debe aparecer la siguiente información:
 - Título, ubicación, precio y superficie de **zulo**
 - Email de contacto del **especulador**
 - Enlace de "Publicar nuevo zulo"
 - Enlace de "Ver más detalles"
 - Crea tu propia vista o modifica la vista proporcionada

[+ Publicar nuevo zulo](#)

Armario empotrado con vistas a mi habitación

Ubicación: Calle Colmena, 21

Superficie: 2 m²

450€ / mes

Contacto: especulador@correo.com

[Ver más detalles](#)

2. Detalle de un zulo (`show`) (**0.75/4**):

- Muestra **todos** los detalles de un zulo
 - Título, ubicación, descripción, precio y superficie
 - También deberás mostrar **todos los detalles** del **especulador**: nombre, apellidos y email
 - Enlace de "Ver más zulos de este especulador" (`/especuladores/{especulador}/zulos`)
 - Enlace de "Volver al listado"
- Crea tu propia vista o modifica la vista proporcionada

[← Volver al listado](#)

Armario empotrado con vistas a mi habitación

Ubicación: Calle Colmena, 21

Descripción:

Espacio optimizado al máximo, ideal para personas de talla baja o contorsionistas

Superficie: 2 m²

450€ / mes

Datos del Especulador

Contacto: Mr. Mentalidad de Tiburón

Email: especulador@correo.com

[Ver más zulos de este especulador](#)

3. Registro de zulo (`create` / `store`) (**1/4**):

- Formulario para crear un nuevo zulo
- Asigna aleatoriamente un id de un especulador. Apóyate en el [Anexo de ayuda](#).

4. Modificación y eliminación de zulo (`edit` / `update` / `destroy`) (**0.5/4**):

- Informa al usuario de que no está implementado con un código HTTP 501

Mapea la ruta `/especuladores/{especulador}/zulos` al método del controlador adecuado para **mostrar una lista de zulos** en la que solamente aparezcan los zulos publicados por ese **especulador**. Deberás implementar este método en el controlador y devolver la vista adecuada (**0.5/4**).

4. Filtrado (2/10)

Aprovecha el formulario proporcionado o crea el tuyo propio en la vista `index` de **Zulo**. El formulario debe permitir filtrar lo siguiente:

- Precio máximo. Con campo de texto
- Email del especulador. Con campo de texto
- Agregado a favoritos por algún usuario. Con checkbox

Normas de entrega

1. Un fichero `ZIP` que contenga la carpeta del proyecto **sin** `node_modules` **ni** `vendor`
2. Un fichero `TXT` que contenga los comandos ejecutados en el **Ejercicio 1**

Anexo de ayuda

Para obtener una entidad aleatoria de clase `Modelo`

```
Modelo::inRandomOrder()->first()
```