



PHP - PR03

Nombre del alumno

Ejercicio entregable (Enunciado y entregable)

Ejercicio #1 LS Inmuebles

La empresa LS Inmuebles nos pide crear una API web, para conectar después con una aplicación cliente, y gestionar sus pisos en alquiler y los trabajadores encargados de gestionarlos.

Estándar de código

Desarrolla la aplicación usando el estándar PSR12. Para cada error en los estándares se restará 0,25 a la nota final de la práctica, hasta un máximo de 2 puntos.

Si encuentras un error que crees que no se debería tener en cuenta, coméntalo con el profesor, antes de la fecha de entrega de la práctica.

Validación

La práctica deberá ser validada con una entrevista si el profesor lo considera conveniente.

Control de versiones

Utiliza el enlace de Github Classroom para crear tu aplicación. El uso de GitHub es obligatorio, con un mínimo de 1 commit (y push) por funcionalidad, con los avances de código correspondientes.

Grupos

Puedes realizar la aplicación solo o en pareja. Si divides la práctica en pareja, se aplicarán los siguientes cambios:

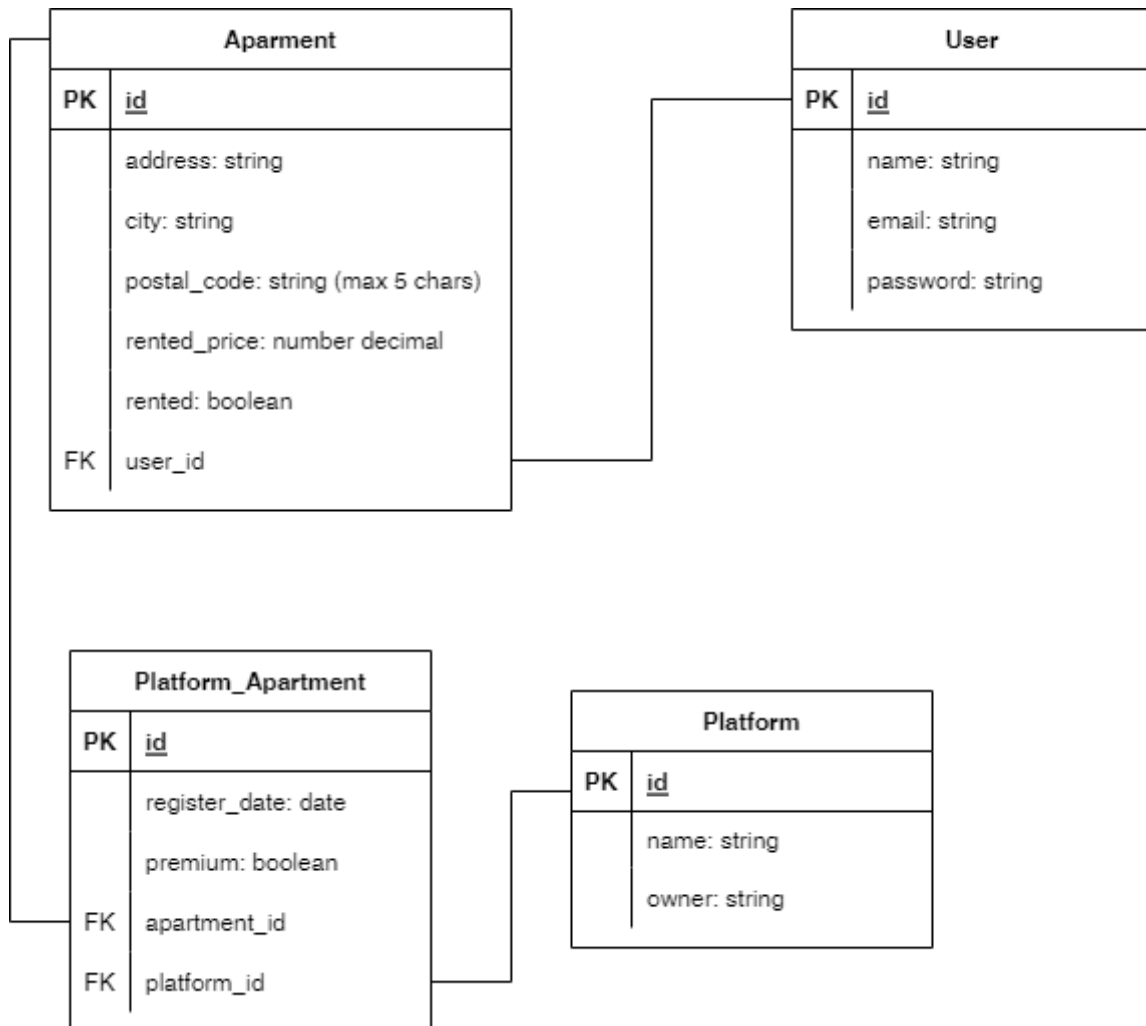
- Debéis realizar mínimo:
 - Una rama para cada developer, a parte de la master.
 - Cada persona deberá usar una cuenta de Git diferente y hacer commits desde su cuenta para poder comprobar el trabajo realizado de cada persona.
 - Un commit por endpoint.
- Si revisando los commits de cada developer el profesor considera que uno no ha realizado suficiente trabajo propio, esta práctica no será válida.

Entregables

- Zip del proyecto de laravel
 - Código
 - Exportación de la BBDD
- Memoria con capturas del Trabajo realizado:
 - Comandos de artisan que se van usando.
 - Una/s captura/s de cada endpoint de la API demostrando su correcto funcionamiento

Migrations:

(1p) La BBDD tendrá el siguiente esquema, créalo usando migrations:



Factories y seeders:

(0.25p) Usa las Factories y Seeders para crear 5 usuarios de testing.

(0.25p) Usa las Factories y Seeders para crear 20 apartamentos iniciales de testing, asígnales los usuarios anteriores.

(0.25p) Usa las Factories y Seeders para crear 3 plataformas.

(0.25p) Usa las Factories y Seeders para crear 50 relaciones entre plataformas y apartments.

Los endpoints de la API deben ser los siguientes:

- (0.5p) Registro:
 - URL: /signin
 - Parámetros validados con uno o varios middlewares:
 - Name: requerido, string, solo letras o espacios.
 - Email: requerido, string y formato de email.
 - Password: requerido, string, mínimo 5 caracteres.
 - Password confirmation: requerido, debe ser igual que el password.
 - Acciones:
 - Deberá guardar el usuario con sus datos a la BBDD.
- (0.5p) Login:
 - URL: /signup
 - Parámetros validados con uno o varios middlewares:
 - Email: requerido, string y formato de email.
 - Password: requerido, string, mínimo 5 caracteres.
 - Acciones:
 - Deberá validar que el usuario con email y password exista y devolver un token de autenticación.
- (0.5p) Crear Apartment:
 - URL: Crea la ruta usando "resource" con la ruta "apartments".
 - Parámetros validados con uno o varios middlewares:
 - Address: requerido, string, máximo 100 caracteres.
 - City: requerido, string.
 - PostalCode: requerido, deben ser 5 caracteres.
 - RentPrice: opcional, debe ser un número positivo, con máximo 2 decimales.
 - Rented: requerido, debe ser un boolean para marcar si está alquilado actualmente o no.
 - Acciones:
 - Con un middleware deberá validar que se reciba un token de autenticación.
 - Se debe crear un "apartment" o piso (recomendamos poner los nombres en inglés) con los campos indicados.
 - El campo "manager_id" se pondrá automáticamente extraído del ID del usuario del token.
- (0.5p) Actualizar Apartment:
 - URL: Crea la ruta usando "resource" con la ruta "apartments".
 - Parámetros validados con uno o varios middlewares:
 - ID: requerido, número.
 - Address: opcional, string, máximo 100 caracteres.
 - City: opcional, string.
 - PostalCode: opcional, deben ser 5 caracteres.
 - RentPrice: opcional, debe ser un número positivo, con máximo 2 decimales.
 - Rented: opcional, debe ser un boolean para marcar si está alquilado actualmente o no.
 - Acciones:
 - Con un middleware deberá validar que se reciba un token de autenticación.
 - Con otro middleware se validará que el usuario del token sea el mismo que está en su campo "manager_id".

- Se debe actualizar un "apartment" o piso (recomendamos poner los nombres en inglés) con los campos indicados.
- (0.5p) Eliminar Apartment:
 - URL: Crea la ruta usando "resource" con la ruta "apartments".
 - Parámetros validados con un middleware:
 - ID: requerido, número.
 - Acciones:
 - Con un middleware deberá validar que se reciba un token de autenticación.
 - Con otro middleware se validará que el usuario del token sea el mismo que está en su campo "manager_id".
 - Se debe eliminar el "apartment" o piso (recomendamos poner los nombres en inglés) indicado.
- (0.5p) Mostrar un Apartment:
 - URL: Crea la ruta usando "resource" con la ruta "apartments".
 - Parámetros validados con un middleware:
 - ID: requerido, número.
 - Acciones:
 - Se debe mostrar todos los campos del "apartment" o piso (recomendamos poner los nombres en inglés) indicado.
 - Se debe mostrar el usuario (manager) que lo gestiona.
 - Se deben mostrar las plataformas en las que está presente.
- (1p) Mostrar todos los Apartments:
 - URL: Crea la ruta usando "resource" con la ruta "apartments".
 - Parámetros sin validación (sin middleware):
 - City: optional, string.
 - Acciones:
 - Se debe mostrar todos los campos de todos los "apartments" o pisos (recomendamos poner los nombres en inglés).
 - Si se envía una Ciudad como parámetro, se deben devolver solo los apartamentos de esa Ciudad.
 - Se deben mostrar los usuarios (managers) que los gestionan.
 - Se deben mostrar las plataformas en las que está presente.
- (1p) Mostrar todos los apartamentos de una plataforma:
 - URL: /platform
 - Parámetros validados (sin middleware):
 - Platform_id: requerido.
 - Acciones:
 - Se deben mostrar los apartamentos de una plataforma en concreto.
 - Se deben mostrar los usuarios (managers) que los gestionan.
- (1p) Mostrar los pisos si están alquilados o no:
 - URL: /apartments_rented
 - Parámetros validados (sin middleware):
 - Rented: requerido, debe ser un boolean para marcar si está alquilado actualmente o no.
 - Acciones:

- Se debe mostrar todos los campos de todos los "apartments" o pisos (recomendamos poner los nombres en inglés) que están o no alquilados.
 - Se deben mostrar los usuarios (managers) que los gestionan.
- (1p) Mostrar los pisos si el precio de alquilar es superior a 1000€:
 - URL: /apartaments_premium
 - Acciones:
 - Se debe mostrar todos los campos de todos los "apartments" o pisos (recomendamos poner los nombres en inglés) que su precio sea superior a 1000.
 - Se deben mostrar los usuarios (managers) que los gestionan.

Requisitos varios:

- La autenticación debe ser usando la librería sanctum.
- (1p) El código se debe escribir con calidad. Esta puntuación solo será válida si con los demás puntos se obtiene un mínimo de 5.