

Prueba Técnica

Requerimiento General:

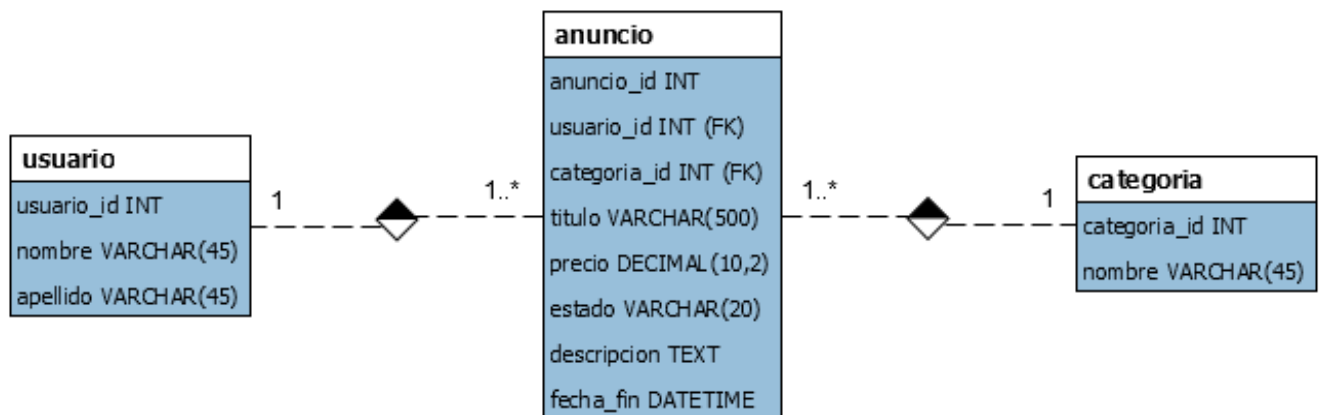
Se debe desarrollar una API para una aplicación móvil. Dentro de esta app los usuarios pueden ver y publicar anuncios con el objetivo final de vender los artículos solicitados. Para poder publicar un anuncio el usuario debe estar logueado, pero no es necesario estar registrado para poder ver todos los anuncios publicados.

Requerimientos Tecnicos

- La API debe ser desarrollada utilizando Laravel, versión 10 o mayor.
- Base de datos MySql

Requerimientos Funcionales:

- Crear tres tablas: *usuario*, *anuncio* y *categoria*. La tabla *categoria* debe tener 3 categorías: Tecnologia, Deportes, Moda



*** Si considera conveniente, los nombres de las columnas, tablas y tipos de datos se pueden modificar, así como también agregar columnas en caso sea necesario.*

- Desarrollar 3 endpoints para manipular y obtener información.
 - **Crear un Anuncio:** Los usuarios logueados pueden crear anuncios, solicitando la siguiente información:
 - Título: título del anuncio, requerido, solo permitir Aa-Zz
 - Precio: precio del artículo a vender, requerido.
 - Estado: estado del artículo, requerido, las opciones a seleccionar son: [nuevo, usado, restaurado, como nuevo]
 - Descripción: descripción del artículo a vender, opcional
 - Fecha Fin: fecha hasta cuando el anuncio estará visible, requerido.
 - Id de la categoría: id de la categoría donde se quiere asignar el anuncio, requerido.
 - **Cancelar un Anuncio:** Los usuarios logueados tienen la opción de borrar/cancelar un anuncio, con esta acción, el anuncio ya no será visible para nadie. Queda a discreción la forma de implementarlo, ya sea borrando el registro, o actualizando algún campo extra en la tabla de anuncios.
 - **Listar Anuncios:** listar todos los anuncios creados. Este endpoint debe ser público, los usuarios no logueados pueden ver los anuncios.
 - Información a retornar:
 - Nombre y Apellido de la persona que creó el anuncio
 - Nombre de la categoría.
 - Id del anuncio, Título, precio, estado y descripción.
 - Fecha en la que el anuncio fue creado.
 - Ejemplo:

```
{
  "usuario": {
    "nombre": "Demo",
    "apellido": "User"
  },
}
```

```
"categoria": {
  "nombre": "Deporte"
},
"anuncio_id": 22323,
"titulo": "Titleist Men's 718 AP1 Irons",
"precio": 58.99,
"estado": "new",
"descripcion": "",
"fecha_creacion": "2024-06-03 08:50:29"
}
```

- Los criterios de búsqueda serían:
 - Por rango de precios
 - Por categoria
 - Estado
 - Palabras claves
 - Filtro Booleano, “Mostrar Todos”
 - Cuando el filtro es **false**: no mostrar anuncios donde la fecha fin sea mayor a la actual y deben ser ordenados del más antiguo al más nuevo.
 - Cuando el filtro es **true**: mostrar todos los anuncios y ordenar por **Precio** de mayor a menor.
- Los anuncios se deben retornar paginados

Puntos a Evaluar

- Estructura del proyecto
- Validación de todos los datos a solicitar
- Manejo de los errores y la respuesta que retorna. (Códigos HTTP, códigos de errores json etc)

- Seguridad a implementar para los endpoints donde es necesario identificar al usuario que quiere crear o cancelar el anuncio (Sanctum, Passport etc)
- Uso de convención de nombres para una API Rest (nombre de Urls, protocolo HTTP para aplicar a cada endpoint, etc)

Forma de Entrega

- Enviar por correo electrónico el link de repositorio de github donde se ha subido el proyecto.
- Incluir en el proyecto:
 - Script de la base de datos con datos de pruebas
 - Collection de postman con los 3 endpoints creados