

## Artículos

Crear una aplicación web simple que permita a los usuarios registrarse, iniciar sesión, listar y publicar artículos en un blog, con o sin imágenes y recibir comentarios en los mismos.

Los requisitos específicos son los siguientes:

La aplicación debe utilizar Laravel 7 con Passport y MySQL para la base de datos.

**No se debe implementar un frontend.** Solo una API Rest (backend) utilizando los correspondientes tokens de autenticación.

Se debe entregar una colección (POSTMAN) con los endpoints pedidos y la dirección de un repositorio privado donde se encontrará la aplicación con los pasos a seguir para realizar el despliegue inicial.

Los usuarios deben poder registrarse con un nombre de usuario, dirección de correo electrónico y una contraseña. La validación debe garantizar que no se permitan nombres de usuario o direcciones de correo electrónico duplicados.

Los usuarios deben poder iniciar sesión con su nombre de usuario y contraseña.

Los usuarios autenticados deben poder ver una lista de todos los artículos publicados en el blog con sus imágenes y comentarios.

Los usuarios autenticados deben poder crear nuevos artículos para el blog, que deben incluir un título y contenido. Los artículos deben estar asociados con el usuario que los creó.

Los usuarios autenticados deben poder agregar o quitar imágenes a un artículo.

Los usuarios autenticados deben poder eliminar los artículos publicados en el blog.

Se deben poder recibir comentarios en un artículo publicado sin necesidad de estar autenticados.

Se debe poder listar a los usuarios con sus estadísticas (cantidad de artículos, cantidad de artículos con imágenes, cantidad de artículos sin imágenes y cantidad de comentarios por artículo).

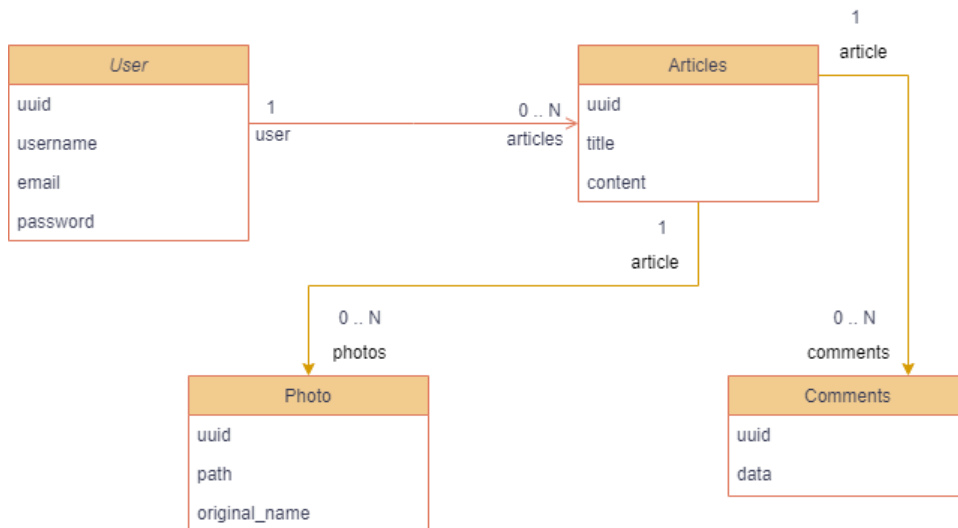
## Multilenguaje:

- El desarrollo debe implementar internacionalización (i18n). O sea, los textos fijos de las pantallas deben estar traducidos, en Inglés y Español.
- El idioma a utilizar puede ser configurable por archivo de configuración (.env) (no es necesario que el idioma se cambie desde la pantalla)
- No son importantes las traducciones en sí, sino cómo se resuelven.

## Recomendaciones:

- Utilizar API Resources (<https://laravel.com/docs/7.x/eloquent-resources>)
- Utilizar Form Request Validation (<https://laravel.com/docs/7.x/validation#form-request-validation>)
- Utilizar Laravel Passport (<https://laravel.com/docs/7.x/passport>)

## Modelo de datos:



### Endpoints esperados:

1. Registrarse: POST /signup
2. Iniciar sesión: POST /signin
3. Listar artículos de un usuario: GET /articles
4. Crear artículos para un usuario: POST /articles
5. Eliminar artículos de un usuario: DELETE /articles/<uuid del artículo>
6. Agregar imágenes a un artículo POST /articles/<uuid del artículo>/photos
7. Eliminar imágenes a un artículo DELETE /articles/<uuid del artículo>/photos/ <uuid de la imagen>
8. Agregar comentarios a un artículo POST /articles/<uuid del artículo>/comments
9. Eliminar comentarios a un artículo DELETE /articles/<uuid del artículo>/ comments/ <uuid del comentario>
10. Listar usuarios con sus respectivas estadísticas GET /statistics

### Entregable:

- Enviar un mail a [scaporale@fnartes.gob.ar](mailto:scaporale@fnartes.gob.ar) con:
  - La URL a un repositorio privado (Github, gitlab, etc.) con el proyecto
    - Agregar como colaboradores:
      - [scaporale@fnartes.gob.ar](mailto:scaporale@fnartes.gob.ar)
      - [isalinas@fnartes.gob.ar](mailto:isalinas@fnartes.gob.ar)
  - Utilizar un archivo readme.md para describir el despliegue del proyecto en un equipo local, utilizando notación markdown.
  - Adjuntar una colección POSTMAN con los endpoints pedidos.
  - Incluir migraciones, seeders y factories para generar información de prueba.