

PROYECTO FIN DE CURSO JAVA FULLSTACK 2020

Borja Díaz Galiana

Ana Sabater Aguado



RESUMEN

El proyecto consta de una aplicación móvil, donde los usuarios pueden visionar, subir, editar y borrar sus fotografías desde su dispositivo móvil. Contiene, además, la función de cambiar entre un tema claro y uno oscuro.

ÍNDICE

Parte Back-end	3
• Petición Get	3
• Petición Post	3
• Petición Put	4
• Petición Delete	4
Back-end adicional	5
• Petición Get	5
• Petición Post	5
• Petición Put	6
• Petición Delete	7
Parte Front-end	8

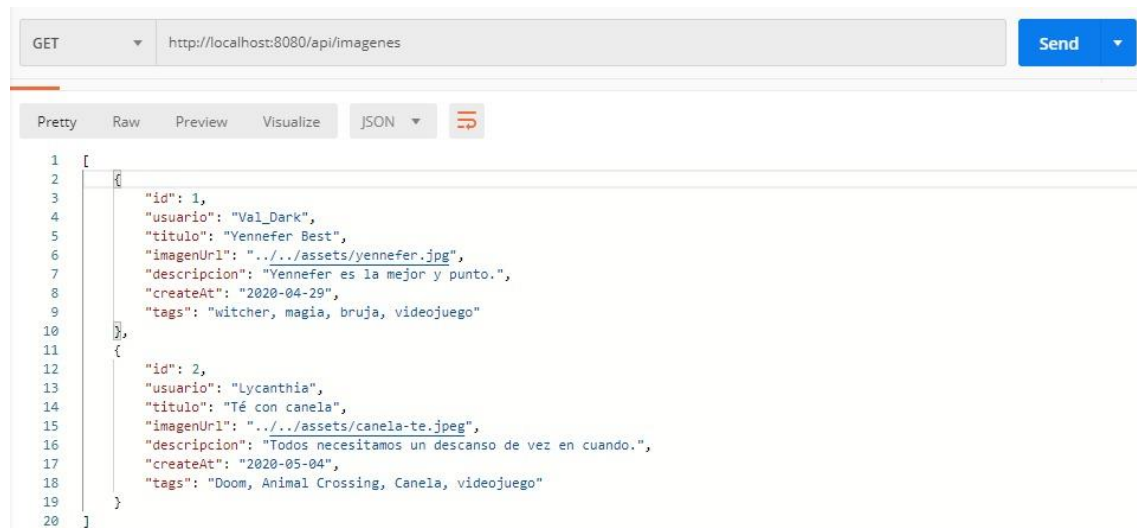
PARTE BACK-END

El proyecto back-end utilizado para la versión final de este proyecto consta de 1 única tabla:

```
18 @Entity
19 @Table(name = "imagenes")
20 public class Imagen implements Serializable {
21
22     @Id
23     @GeneratedValue(strategy = GenerationType.IDENTITY)
24     private Long id;
25     @Size(min=4,max=12,message="El usuario debe tener entre 4 y 12 caracteres.")
26     @NotEmpty(message="Introduce el usuario.")
27     @Column(nullable=false)
28     private String usuario;
29     @NotEmpty(message="Introduce un título.")
30     @Column(nullable=false)
31     private String titulo;
32     @NotEmpty(message="Introduce una imagen.")
33     @Column(nullable=false, name = "imagen_url", columnDefinition="TEXT")
34     private String imagenUrl;
35     private String descripcion;
36     @Column(nullable=false, name = "create_at")
37     @Temporal(TemporalType.DATE)
38     private Date createAt;
39     private String tags;
```

- **Petición GET**

Permite recuperar todos los datos pertenecientes a la correspondiente tabla imágenes.



- **Petición POST**

Permite añadir una imagen, rellenando todos los campos que en la base de datos sean NOT NULL (es decir, que sea obligatorio proporcionarles un valor). Nos devuelve la nueva imagen introducida.

POST http://localhost:8080/api/imagenes Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON**

```

1 {
2   "usuario": "Lycanthia",
3   "titulo": "Descanso",
4   "imagenUrl": "https://drive.google.com/uc?export=view&id=1URg_thEKu5S1N-Uow9tDQcCWoguV3beS",
5   "descripcion": "Todos necesitamos un descanso de vez en cuando.",
6   "createAt": "2020-05-04"
7 }

```

- **Petición PUT**

Permite actualizar una imagen a partir de su id, y nos devuelve la imagen ya actualizada.

PUT http://localhost:8080/api/imagenes/3 Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON**

```

1 {
2   "usuario": "Borjamatico",
3   "titulo": "Descanso",
4   "imagenUrl": "https://drive.google.com/uc?export=view&id=1URg_thEKu5S1N-Uow9tDQcCWoguV3beS",
5   "createAt": "2020-05-04"
6 }

```

GET http://localhost:8080/api/imagenes/3 Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 42 ms Size: 451 B Sav

Pretty Raw Preview Visualize **JSON**

```

1 {
2   "id": 3,
3   "usuario": "Borjamatico",
4   "titulo": "Descanso",
5   "imagenUrl": "https://drive.google.com/uc?export=view&id=1URg_thEKu5S1N-Uow9tDQcCWoguV3beS",
6   "descripcion": null,
7   "createAt": "2020-05-04",
8   "tags": null
9 }

```

- **Petición DELETE**

Nos permite borrar una imagen de la base de datos a partir de su id.

DELETE http://localhost:8080/api/imagenes/3 Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 146 ms Size: 301 B Sav

Pretty Raw Preview Visualize **JSON**

```

1 {
2   "mensaje": "La imagen se borró correctamente."
3 }

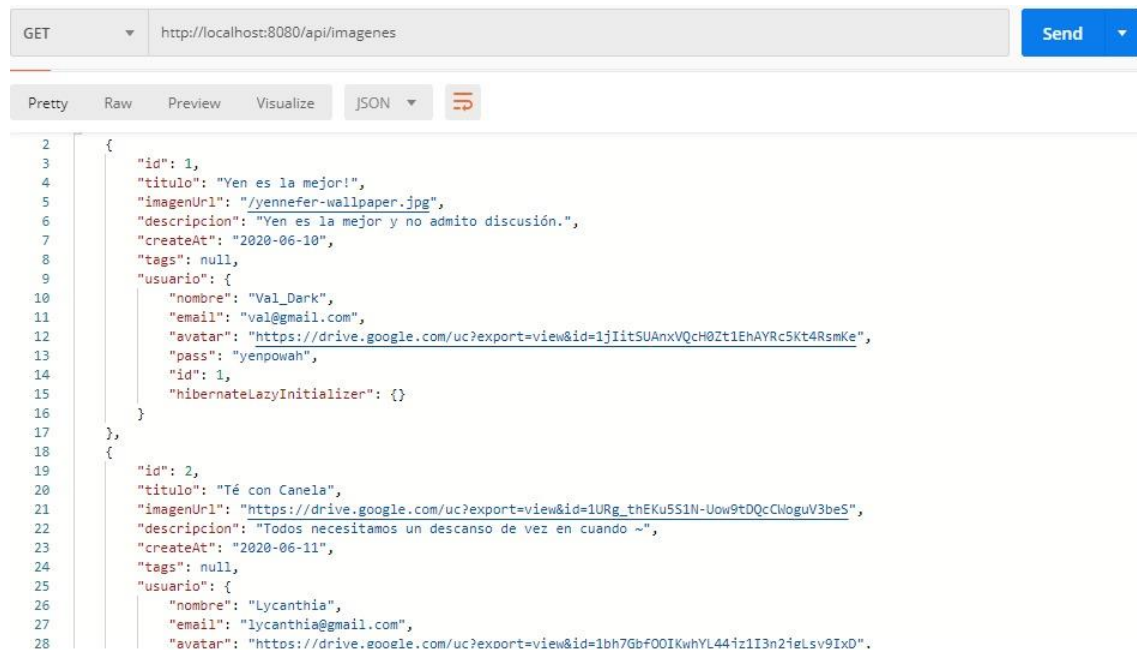
```

BACK-END ADICIONAL

Presentamos, además, un proyecto back-end preparado para tratar con login y registro de usuarios. Finalmente no forma parte de la aplicación definitiva, porque nos fue imposible acabar el front-end con estas funciones.

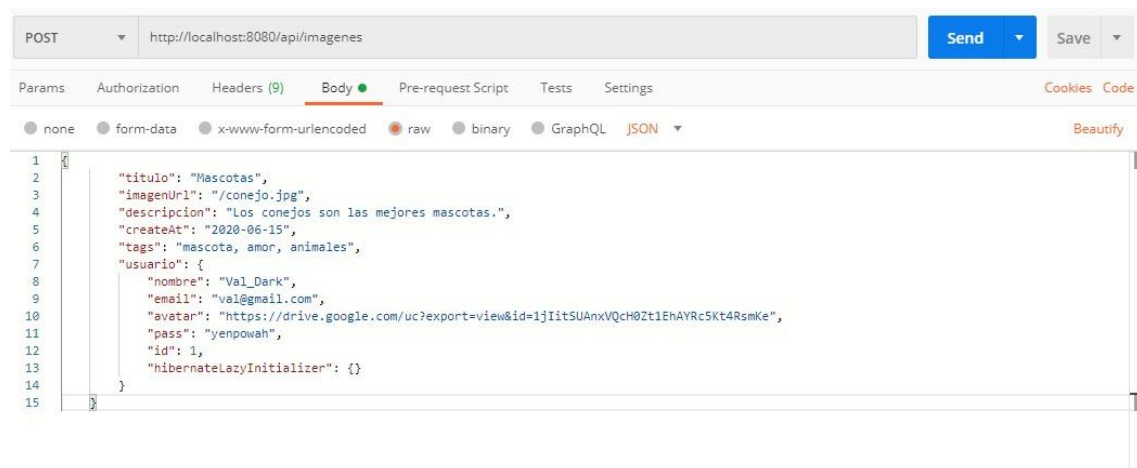
- **Petición Get**

Mediante un GET a la dirección indicada en la captura, recuperamos los datos de la tabla imagenes. Esta tabla tiene una relación Muchos a Uno (Many To One) con la tabla usuarios.



- **Petición Post**

Permite añadir una imagen.



Esta petición devuelve la imagen añadida.

```
Body Cookies Headers (8) Test Results Status: 201 Created Time: 58 ms Size: 671 B

Pretty Raw Preview Visualize JSON

1 {
2   "imagen": {
3     "id": 5,
4     "titulo": "Mascotas",
5     "imagenUrl": "/conejo.jpg",
6     "descripcion": "Los conejos son las mejores mascotas.",
7     "createAt": "2020-06-15T00:00:00.000+00:00",
8     "tags": "mascota, amor, animales",
9     "usuario": {
10      "nombre": "Val_Dark",
11      "email": "val@gmail.com",
12      "avatar": "https://drive.google.com/uc?export=view&id=1jIitSUAnxVQcH0Zt1EhAYRc5Kt4RsmKe",
13      "pass": "yempowah",
14      "id": 1
15    }
16  },
17  "mensaje": "La imagen se insertó correctamente."
18 }
```

- **Petición Put**

Permite actualizar una imagen, a partir de su id.

```
PUT http://localhost:8080/api/imagenes/5 Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

1 {
2   "titulo": "PETS",
3   "imagenUrl": "/conejo.jpg",
4   "descripcion": "Los conejos son las mejores mascotas.",
5   "createAt": "2020-06-15",
6   "tags": "mascota, amor, animales",
7   "usuario": {
8     "nombre": "Val_Dark",
9     "email": "val@gmail.com",
10    "avatar": "https://drive.google.com/uc?export=view&id=1jIitSUAnxVQcH0Zt1EhAYRc5Kt4RsmKe",
11    "pass": "yempowah",
12    "id": 1,
13    "hibernateLazyInitializer": {}
14  }
15 }
```

La petición nos devuelve el objeto actualizado.

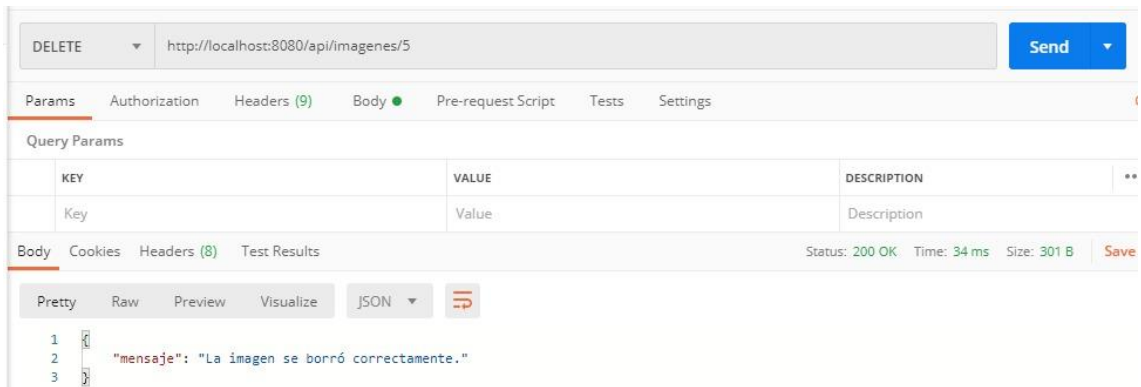
```
Body Cookies Headers (8) Test Results Status: 200 OK Time: 39 ms Size: 648 B Save Response

Pretty Raw Preview Visualize JSON

1 {
2   "imagen": {
3     "id": 5,
4     "titulo": "PETS",
5     "imagenUrl": "/conejo.jpg",
6     "descripcion": "Los conejos son las mejores mascotas.",
7     "createAt": "2020-06-15",
8     "tags": "mascota, amor, animales",
9     "usuario": {
10      "nombre": "Val_Dark",
11      "email": "val@gmail.com",
12      "avatar": "https://drive.google.com/uc?export=view&id=1jIitSUAnxVQcH0Zt1EhAYRc5Kt4RsmKe",
13      "pass": "yempowah",
14      "id": 1
15    }
16  },
17  "mensaje": "La imagen ha sido actualizada con éxito."
18 }
```

- **Petición Delete**

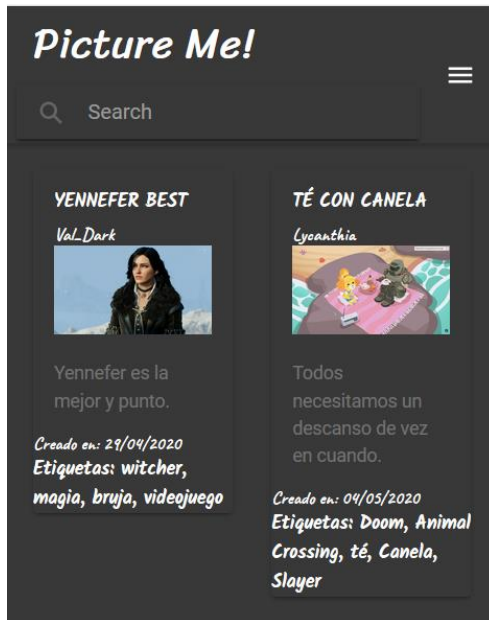
Permite eliminar una imagen de la base de datos, a partir de su id.



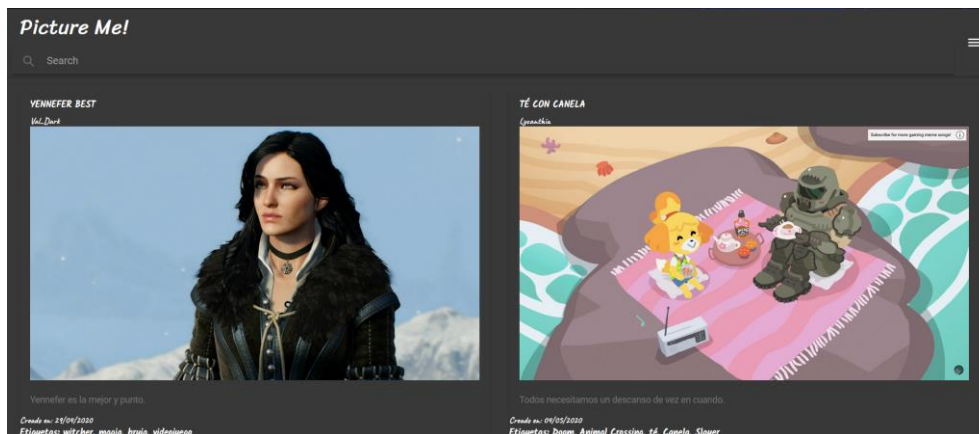
PARTE FRONT

En la parte front, se enseñarán en los modos móvil y escritorio y habrá una breve explicación de cada pantalla.

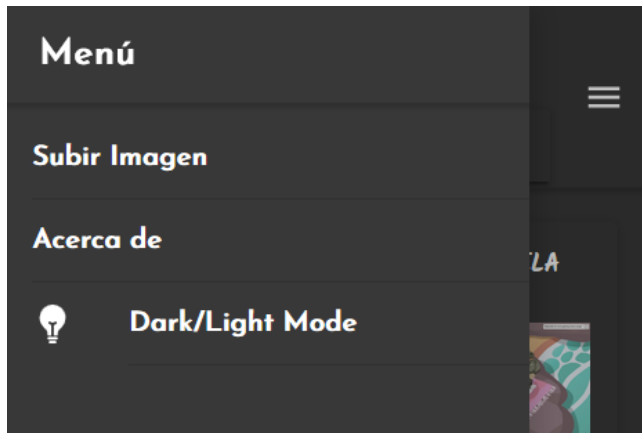
Pantalla Home (modo móvil):



Pantalla Home (modo pc):

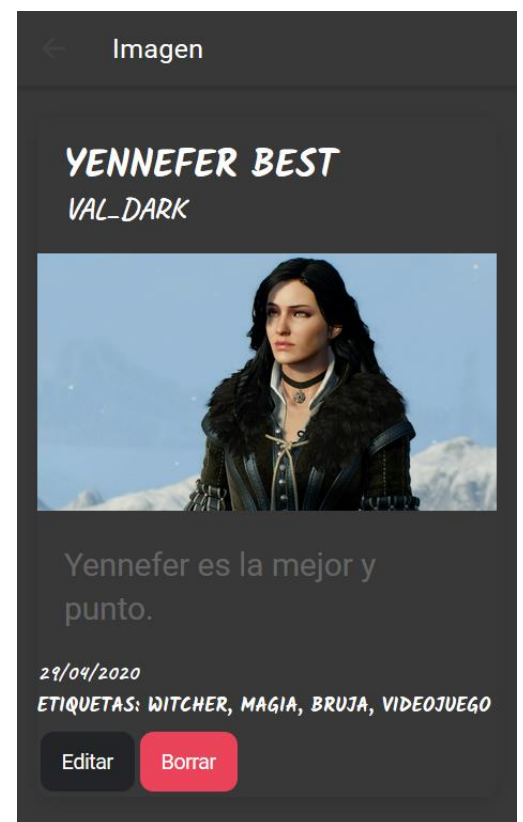
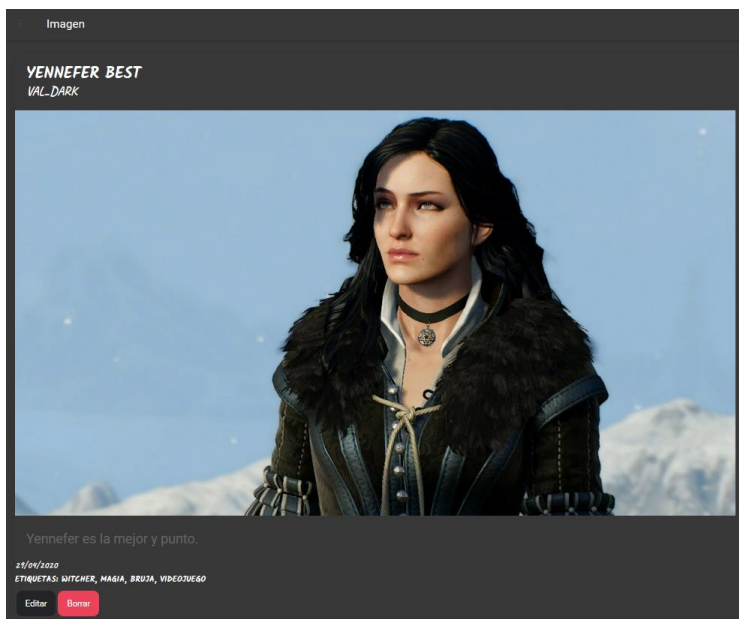


En el home podemos observar el nombre de la aplicación, las imágenes subidas por los usuarios, así de como una barra de búsqueda en la que podemos buscar tanto por nombre de usuario, como por título o etiquetas. También vemos un botón a la parte derecha, donde al pulsarlo, se muestra el siguiente menú desplegable:



Desde este menú, podemos subir una imagen, ver la información de la aplicación y cambiar de una interfaz con colores claros, a una con colores oscuros para poder descansar la vista cuando haya poca luz.

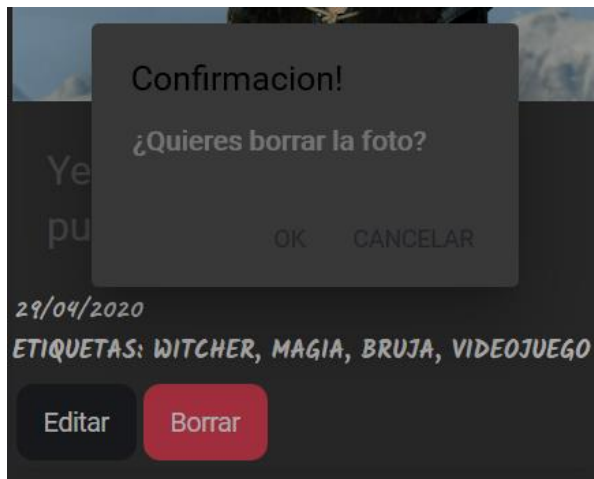
Mostrar Imagen:



Al pulsar en el home, encima de una foto, nombre de usuario, o título, nos lleva a esta página, donde tenemos la opción de borrar, o editar dicha foto.

Al pulsar el botón borrar, se nos muestra esta ventana, donde podemos confirmar o cancelar la acción.

Al pulsar el botón borrar, se nos muestra esta ventana, donde podemos confirmar o cancelar la acción:



Al pulsar el botón editar, nos lleva a esta pantalla, donde podremos editar el título, la descripción y las etiquetas.

← Editar Imagen	← Editar Imagen
Título: Té con canela	Título: Té con canela
Descripción: Todos necesitamos un descanso	Descripción: Todos necesitamos un descanso de vez en cuando.
Etiquetas:	Etiquetas:
ACTUALIZAR	ACTUALIZAR

En la pantalla para subir imagen, se deben rellenar todos los campos, a excepción si el usuario lo considera oportuno, de descripción y etiquetas (campos que en la base de datos pueden ser NULL). La fecha de creación se auto-gestiona añadiendo la fecha actual en la que el usuario ha llevado a cabo la subida. Para añadir una imagen, al darle a examinar se nos abrirá el explorador del sistema para seleccionarla.


← Subir Imagen	← Subir Imagen
Usuario:	Usuario:
Título:	Título:
ImagenUrl: Examinar... No se ha seleccionado ningún archivo.	ImagenUrl: Examinar... No se ha seleccionado ningún archivo.
Descripción:	Descripción:
Etiquetas:	Etiquetas:
AÑADIR ESTA IMAGEN	AÑADIR ESTA IMAGEN

Aquí se puede ver la información de la aplicación, así de como unos botones para ir a los perfiles de LinkedIn y Github, con sus correspondientes iconos.


←


Realizado por:





Borja Díaz ...






Ana Sabat...


Trabajo final del curso de Java Fullstack, impartido por la Escuela de Organización Industrial.

Hemos desarrollado una aplicación móvil que permite a los usuarios compartir fotografías. Estas fotografías pueden además, modificarse o eliminarse por parte del usuario que la subió a la plataforma originalmente.


Tecnologías utilizadas:


← Realizado por:





Borja Díaz Galiana





Ana Sabater Aguado

Trabajo final del curso de Java Fullstack, impartido por la Escuela de Organización Industrial.

Hemos desarrollado una aplicación móvil que permite a los usuarios compartir fotografías. Estas fotografías pueden además, modificarse o eliminarse por parte del usuario que la subió a la plataforma originalmente.

Tecnologías utilizadas:

El front-end se ha realizado con Ionic y Angular. El back-end se ha llevado a cabo con Spring Boot.

Disponible en [Github](#).

11