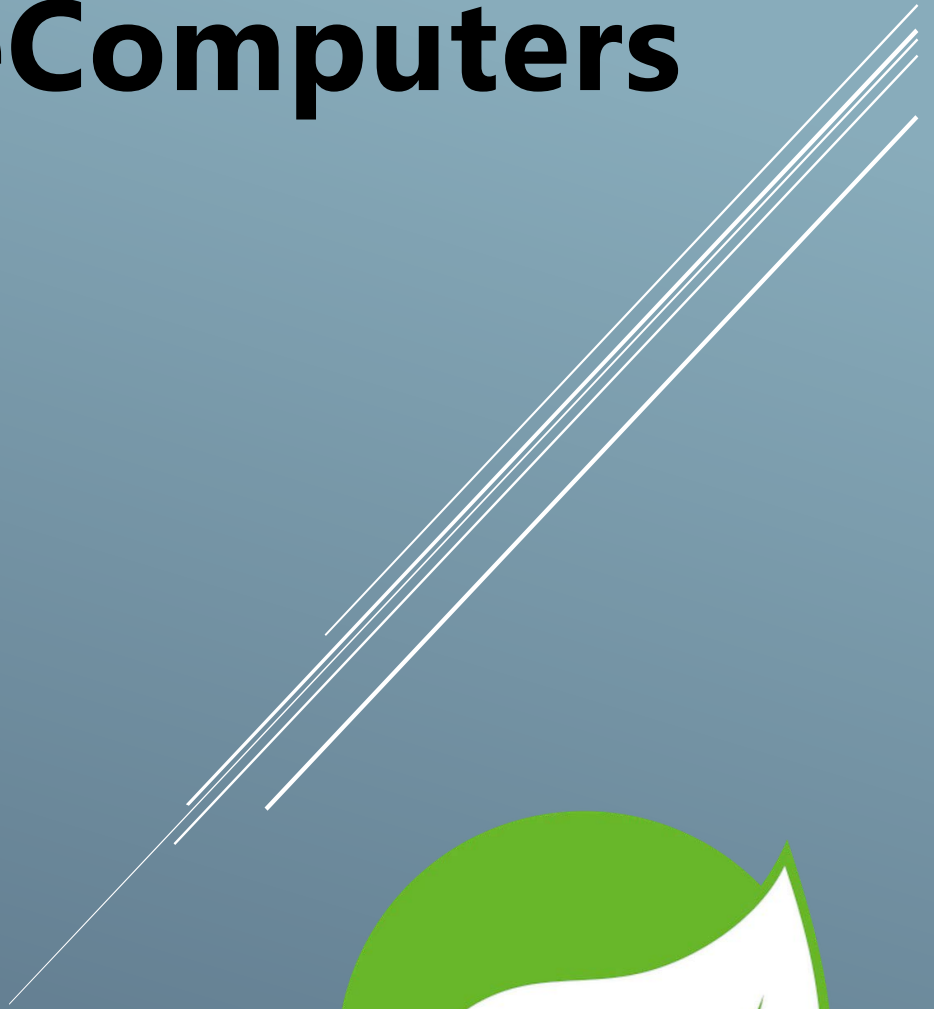


MikeComputers



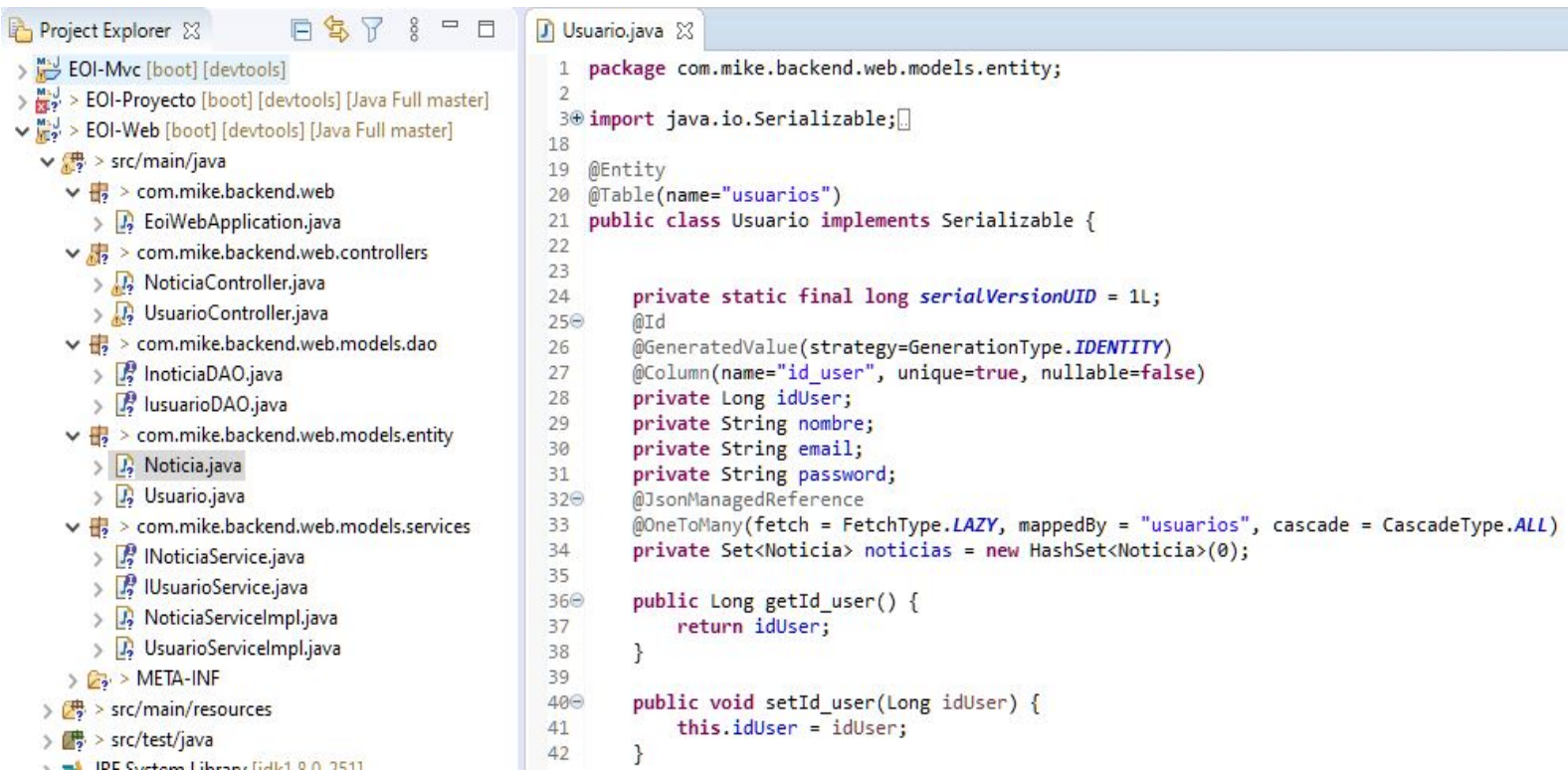
Michael López Martínez

Java full stack

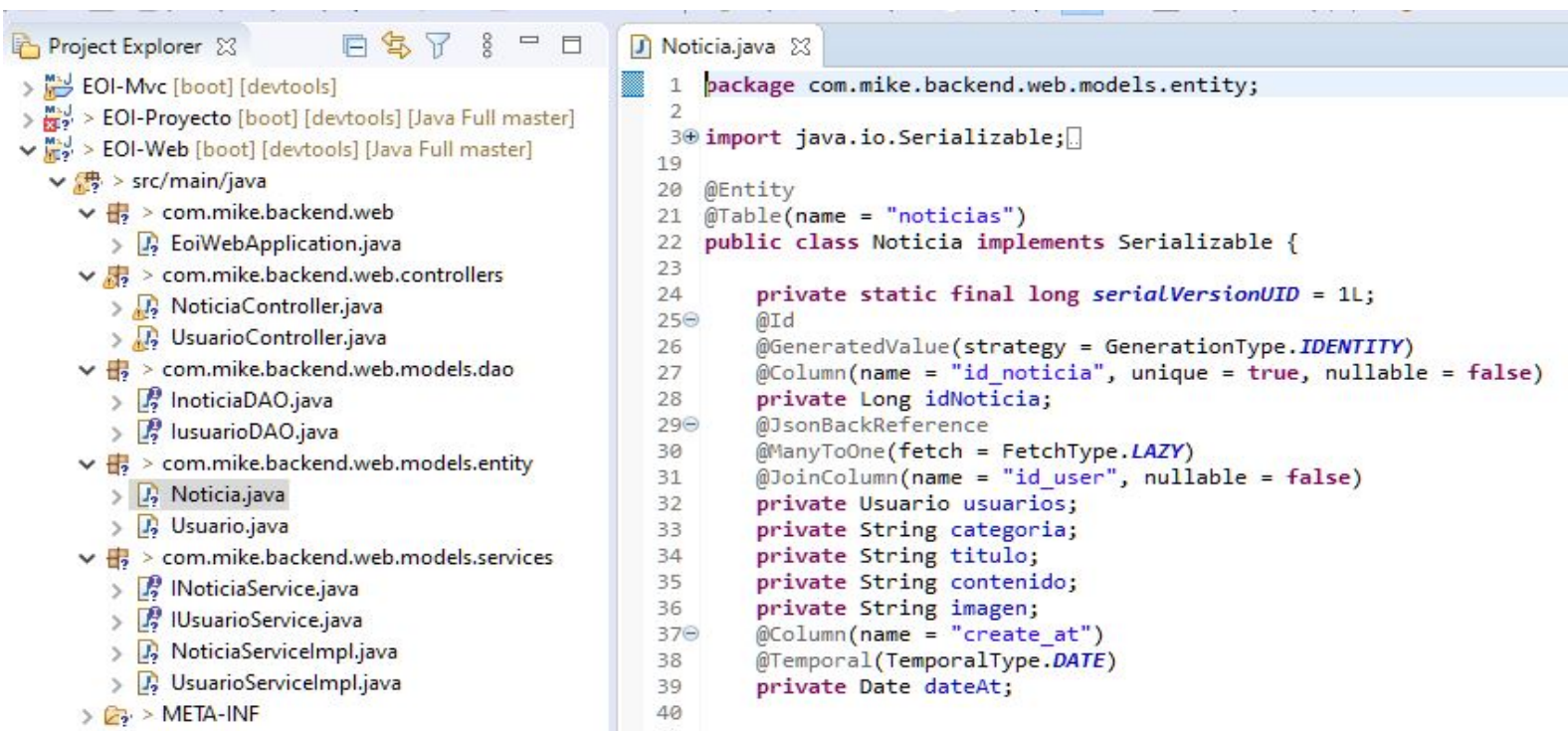
Este proyecto consiste en una pagina web de noticias sobre actualidad tecnológica.

Parte **backend**, creada con Java y Spring.

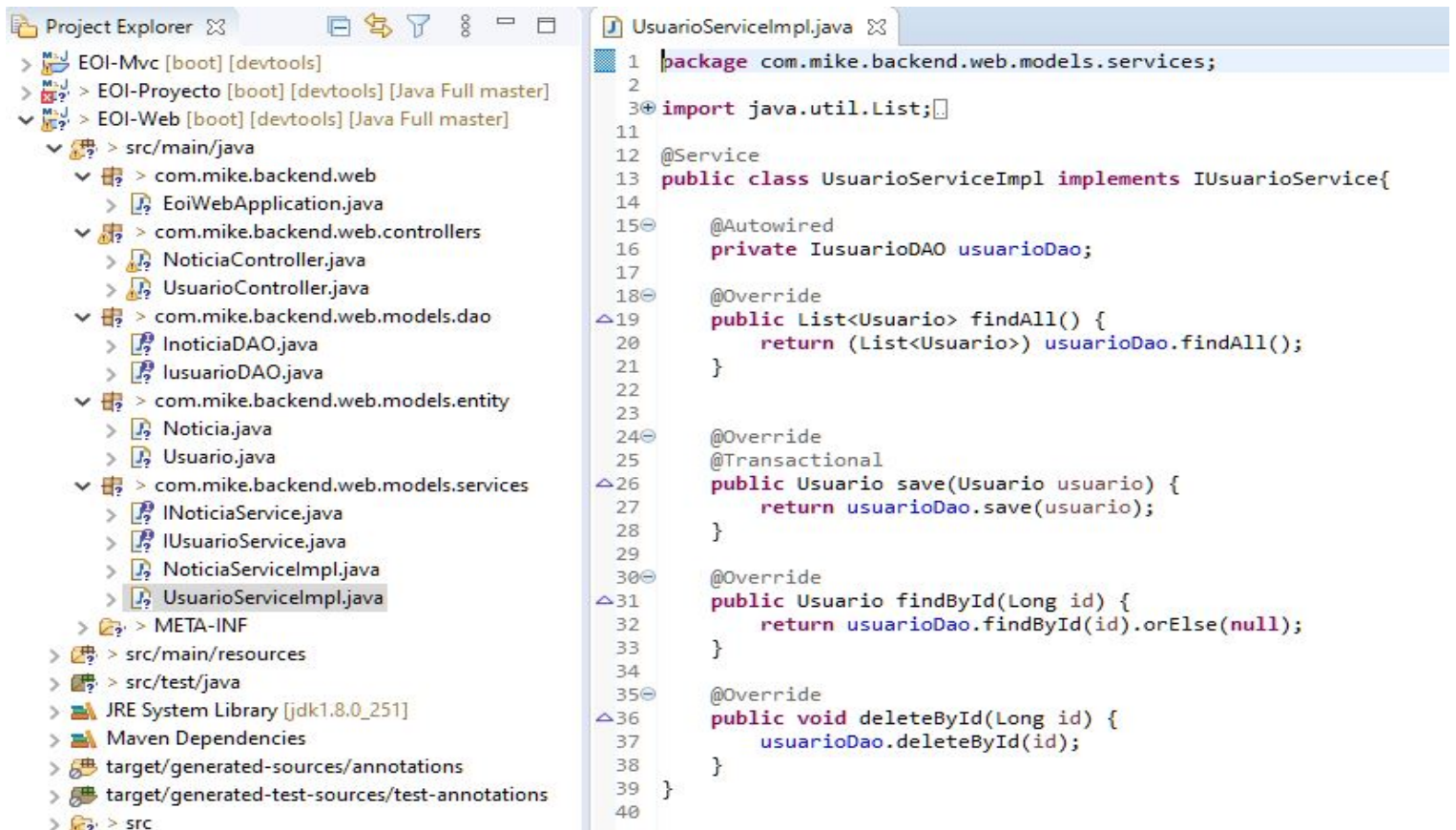
Entidad Usuario:



Entidad Noticia:



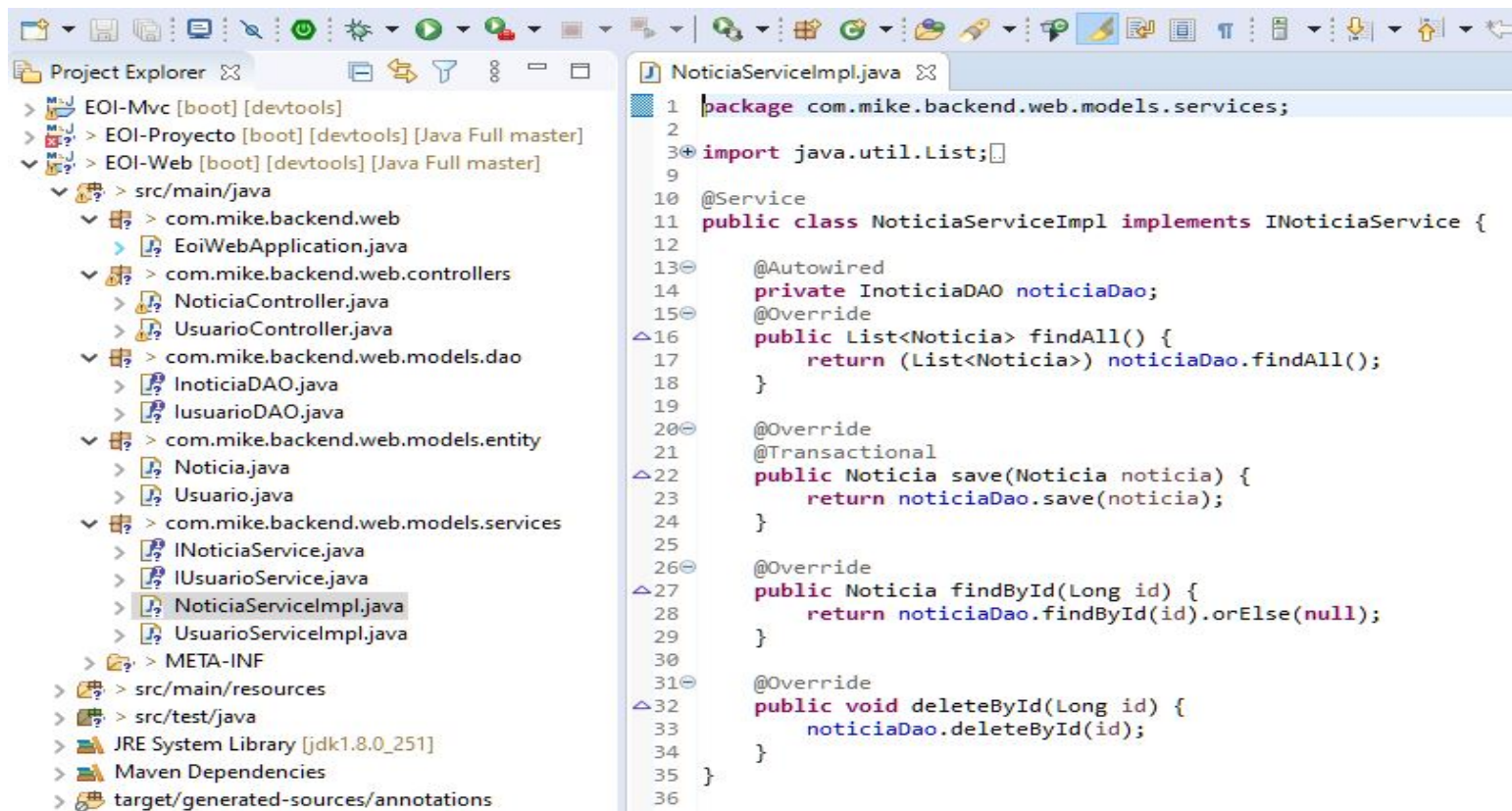
Servicio Usuario:



The screenshot shows an IDE with the Project Explorer on the left and the code editor on the right. The Project Explorer displays the project structure for 'EOI-Mvc', 'EOI-Proyecto', and 'EOI-Web'. The 'src/main/java' directory is expanded, showing the package 'com.mike.backend.web.models.services'. The code editor displays the implementation of the 'UsuarioServiceImpl' class, which implements the 'IUsuarioService' interface. The code includes imports for 'java.util.List' and 'javax.transaction.Transactional', and uses annotations like '@Service', '@Autowired', and '@Override'.

```
1 package com.mike.backend.web.models.services;
2
3 import java.util.List;
4
5 @Service
6 public class UsuarioServiceImpl implements IUsuarioService {
7
8     @Autowired
9     private IusuarioDAO usuarioDao;
10
11     @Override
12     public List<Usuario> findAll() {
13         return (List<Usuario>) usuarioDao.findAll();
14     }
15
16     @Override
17     @Transactional
18     public Usuario save(Usuario usuario) {
19         return usuarioDao.save(usuario);
20     }
21
22     @Override
23     public Usuario findById(Long id) {
24         return usuarioDao.findById(id).orElse(null);
25     }
26
27     @Override
28     public void deleteById(Long id) {
29         usuarioDao.deleteById(id);
30     }
31 }
```

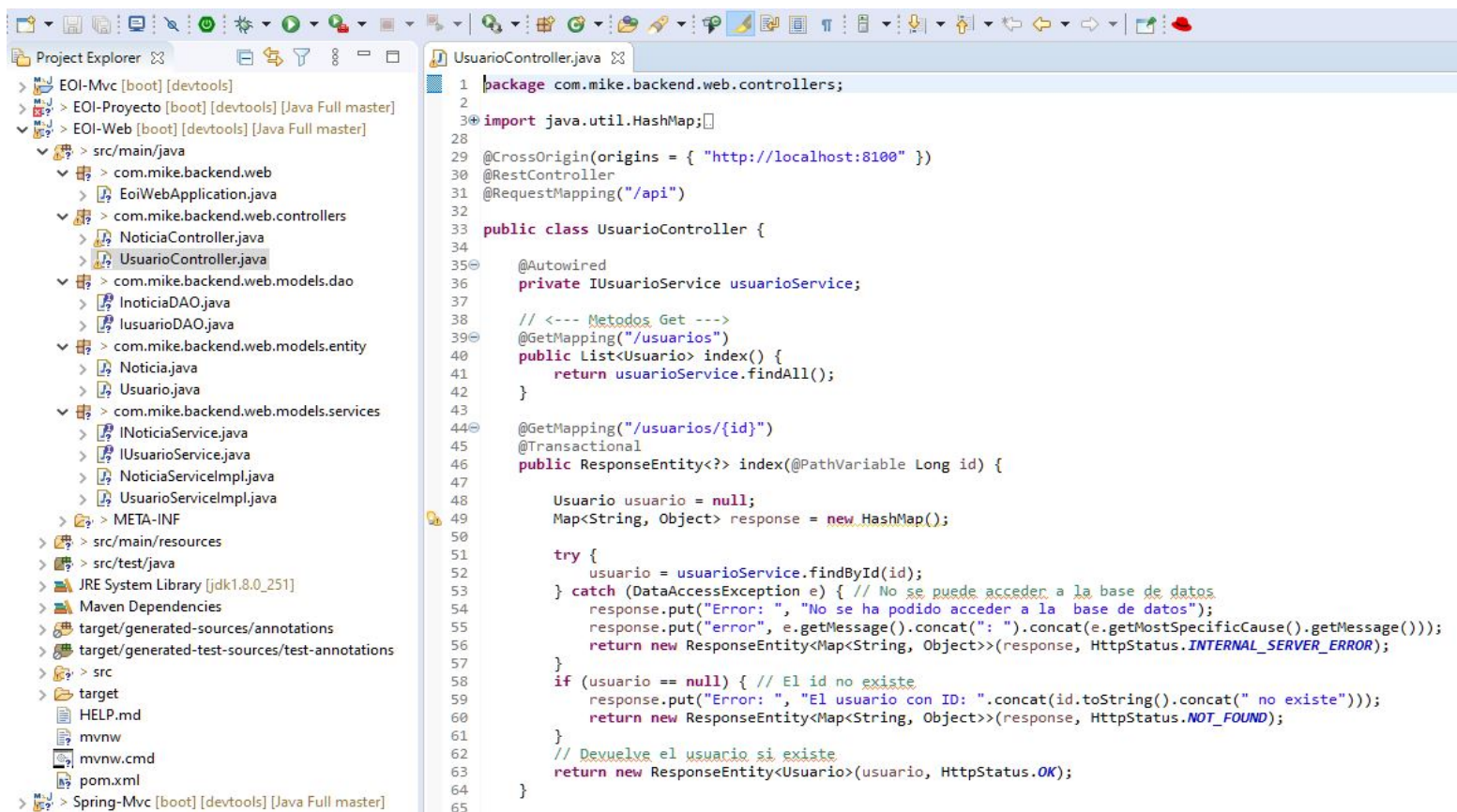
Servicio Noticia:



The screenshot shows an IDE with the Project Explorer on the left and the code editor on the right. The Project Explorer displays the project structure for 'EOI-Mvc', 'EOI-Proyecto', and 'EOI-Web'. The 'src/main/java' directory is expanded, showing the package 'com.mike.backend.web.models.services'. The code editor displays the implementation of the 'NoticiaServiceImpl' class, which implements the 'INoticiaService' interface. The code includes imports for 'java.util.List' and 'javax.transaction.Transactional', and uses annotations like '@Service', '@Autowired', and '@Override'.

```
1 package com.mike.backend.web.models.services;
2
3 import java.util.List;
4
5 @Service
6 public class NoticiaServiceImpl implements INoticiaService {
7
8     @Autowired
9     private INoticiaDAO noticiaDao;
10
11     @Override
12     public List<Noticia> findAll() {
13         return (List<Noticia>) noticiaDao.findAll();
14     }
15
16     @Override
17     @Transactional
18     public Noticia save(Noticia noticia) {
19         return noticiaDao.save(noticia);
20     }
21
22     @Override
23     public Noticia findById(Long id) {
24         return noticiaDao.findById(id).orElse(null);
25     }
26
27     @Override
28     public void deleteById(Long id) {
29         noticiaDao.deleteById(id);
30     }
31 }
```

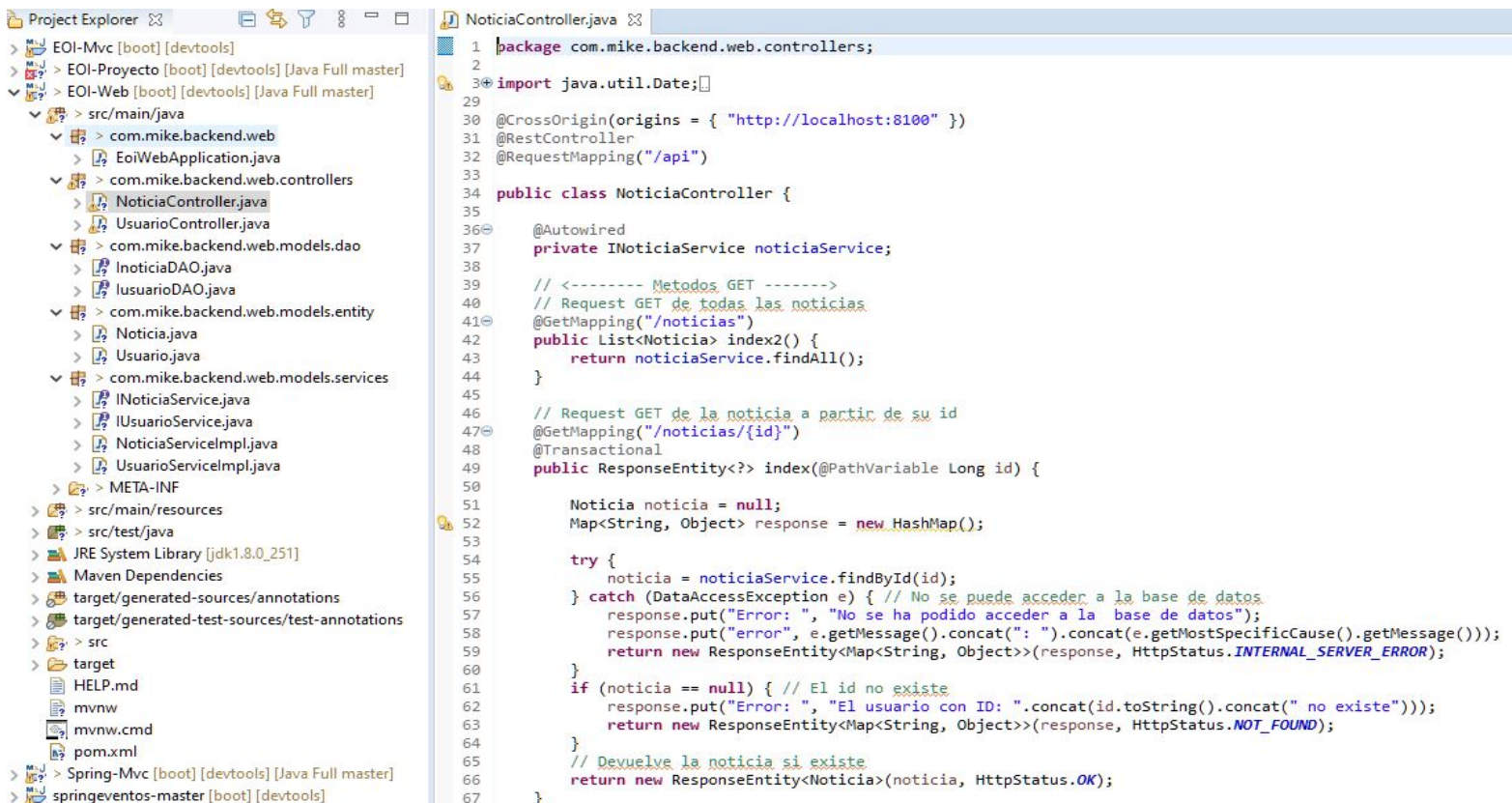
Controller de Usuario:



The screenshot shows an IDE with the Project Explorer on the left and the UsuarioController.java file open in the editor. The Project Explorer shows a project named 'EOI-Web' with a package structure: 'com.mike.backend.web' containing 'controllers', 'models.dao', 'models.entity', and 'models.services'. The 'controllers' package contains 'NoticiaController.java' and 'UsuarioController.java'. The 'models.entity' package contains 'Noticia.java' and 'Usuario.java'. The 'models.services' package contains 'INoticiaService.java', 'IUsuarioService.java', 'NoticiaServiceImpl.java', and 'UsuarioServiceImpl.java'. The 'UsuarioController.java' file is open in the editor, showing the following code:

```
1 package com.mike.backend.web.controllers;
2
3 import java.util.HashMap;
4
5 @CrossOrigin(origins = { "http://localhost:8100" })
6 @RestController
7 @RequestMapping("/api")
8
9 public class UsuarioController {
10
11     @Autowired
12     private IUsuarioService usuarioService;
13
14     // <--- Metodos Get --->
15     @GetMapping("/usuarios")
16     public List<Usuario> index() {
17         return usuarioService.findAll();
18     }
19
20     @GetMapping("/usuarios/{id}")
21     @Transactional
22     public ResponseEntity<?> index(@PathVariable Long id) {
23
24         Usuario usuario = null;
25         Map<String, Object> response = new HashMap();
26
27         try {
28             usuario = usuarioService.findById(id);
29         } catch (DataAccessException e) { // No se puede acceder a la base de datos
30             response.put("Error: ", "No se ha podido acceder a la base de datos");
31             response.put("error", e.getMessage().concat(": ").concat(e.getMostSpecificCause().getMessage()));
32             return new ResponseEntity<Map<String, Object>>(response, HttpStatus.INTERNAL_SERVER_ERROR);
33         }
34
35         if (usuario == null) { // El id no existe
36             response.put("Error: ", "El usuario con ID: ".concat(id.toString()).concat(" no existe"));
37             return new ResponseEntity<Map<String, Object>>(response, HttpStatus.NOT_FOUND);
38         }
39
40         // Devuelve el usuario si existe
41         return new ResponseEntity<Usuario>(usuario, HttpStatus.OK);
42     }
43 }
```

Controller de Noticia:



The screenshot shows an IDE with the Project Explorer on the left and the NoticiaController.java file open in the editor. The Project Explorer shows a project named 'EOI-Web' with a package structure: 'com.mike.backend.web' containing 'controllers', 'models.dao', 'models.entity', and 'models.services'. The 'controllers' package contains 'NoticiaController.java' and 'UsuarioController.java'. The 'models.entity' package contains 'Noticia.java' and 'Usuario.java'. The 'models.services' package contains 'INoticiaService.java', 'IUsuarioService.java', 'NoticiaServiceImpl.java', and 'UsuarioServiceImpl.java'. The 'NoticiaController.java' file is open in the editor, showing the following code:

```
1 package com.mike.backend.web.controllers;
2
3 import java.util.Date;
4
5 @CrossOrigin(origins = { "http://localhost:8100" })
6 @RestController
7 @RequestMapping("/api")
8
9 public class NoticiaController {
10
11     @Autowired
12     private INoticiaService noticiaService;
13
14     // <----- Metodos GET ----->
15     // Request GET de todas las noticias
16     @GetMapping("/noticias")
17     public List<Noticia> index2() {
18         return noticiaService.findAll();
19     }
20
21     // Request GET de la noticia a partir de su id
22     @GetMapping("/noticias/{id}")
23     @Transactional
24     public ResponseEntity<?> index(@PathVariable Long id) {
25
26         Noticia noticia = null;
27         Map<String, Object> response = new HashMap();
28
29         try {
30             noticia = noticiaService.findById(id);
31         } catch (DataAccessException e) { // No se puede acceder a la base de datos
32             response.put("Error: ", "No se ha podido acceder a la base de datos");
33             response.put("error", e.getMessage().concat(": ").concat(e.getMostSpecificCause().getMessage()));
34             return new ResponseEntity<Map<String, Object>>(response, HttpStatus.INTERNAL_SERVER_ERROR);
35         }
36
37         if (noticia == null) { // El id no existe
38             response.put("Error: ", "El usuario con ID: ".concat(id.toString()).concat(" no existe"));
39             return new ResponseEntity<Map<String, Object>>(response, HttpStatus.NOT_FOUND);
40         }
41
42         // Devuelve la noticia si existe
43         return new ResponseEntity<Noticia>(noticia, HttpStatus.OK);
44     }
45 }
```


Pruebas de petición utilizando Postman.
GET de usuario por id.

GET http://localhost:8080/api/usuarios/1

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "nombre": "Michael",
3   "email": "michael@gmail.com",
4   "password": "Michael",
5   "noticias": [
6     {
7       "idNoticia": 12,
8       "categoria": "smartphones",
9       "titulo": "Filtradas las especificaciones de los Redmi 9, Redmi 9A y Redmi 9C",
10      "contenido": "\nCon los Redmi Note 9 ya en la calle, Xiaomi se prepara para otro triple lanzamiento, los Redmi 9, Redmi 9A y R
          batería de 5000 mAh de capacidad, una simple cámara selfie de 5 megapíxeles, conector jack de 3,5mm, puese USB Type-C y ra
          9 llega con una resolución Full HD+ de 2400 x 1080 píxeles propulsado por un SoC MediaTek Helio G70 (2x Cortex-A75 @ 2.00
          64 GB de almacenamiento interno junto a una configuración de cuádruple cámara trasera de 13 + 8 + 5 + 2 megapíxeles. Su pr
          empezar, tendremos una variante con/sin NFC. Ambos comparten un panel IPS HD+ de 1600 x 720 píxeles, son propulsados por u
          almacenamiento interno. En lo que respecta al apartado gráfico, el modelo NFC incluye doble cámara trasera de 13 + 2 megap
          entre los 130 y 150 euros.\nPor último, el Redmi 9A. El modelo más básico mantiene la resolución HD+ pero es propulsado po
          interno. Por otro lado, una única cámara trasera de 13 megapíxeles y un precio que rondará los 100-120 euros\n",
11      "imagen": "https://elchapuzasinformatico.com/wp-content/uploads/2020/06/especificaciones-Redmi-9-Redmi-9A-y-Redmi-9C.jpg",
12      "dateAt": "2020-06-01"
13    },
14    {
15      "idNoticia": 5,
16      "categoria": "informatica",
17      "titulo": "editado",
18      "contenido": "asd",
19      "imagen": "https://elchapuzasinformatico.com/wp-content/uploads/2020/06/Star-Wars-Squadrons-740x416.jpg",
20      "dateAt": "1970-01-01"
21    }
22  ],
23   "id_user": 1
24 }
```

GET de noticia por id.

GET http://localhost:8080/api/noticias/10

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "idNoticia": 10,
3   "categoria": "games",
4   "titulo": "The Last of Us 2 tendrá parche del día uno con modo foto y más novedades",
5   "contenido": "Apenas 10 días nos faltan para poder disfrutar de The Last of Us Parte II. La esperada nueva entrega de 1
          hemos visto las ediciones del juego y sus incentivos de reserva, pero hoy se ha mostrado el contenido del parche de
          posibilidad de visualizar los modelos del juego y una galería de arte conceptual, entre otras funciones",
6   "imagen": "https://il1b.3djuegos.com/juegos/14236/the_last_of_us_2/fotos/maestras/the_last_of_us_2-5123348.jpg",
7   "dateAt": "2020-06-09"
8 }
```

POST de usuario.

POST

http://localhost:8080/api/usuarios

Params

Authorization

Headers (8)

Body ●

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON ▼

```
1
2 {
3     "nombre": "prueba4",
4     "email": "prueba4@gmail.com",
5     "password": "prueba4"
6 }
7
8
```

Body

Cookies

Headers (8)

Test Results

Pretty

Raw

Preview

Visualize

JSON ▼

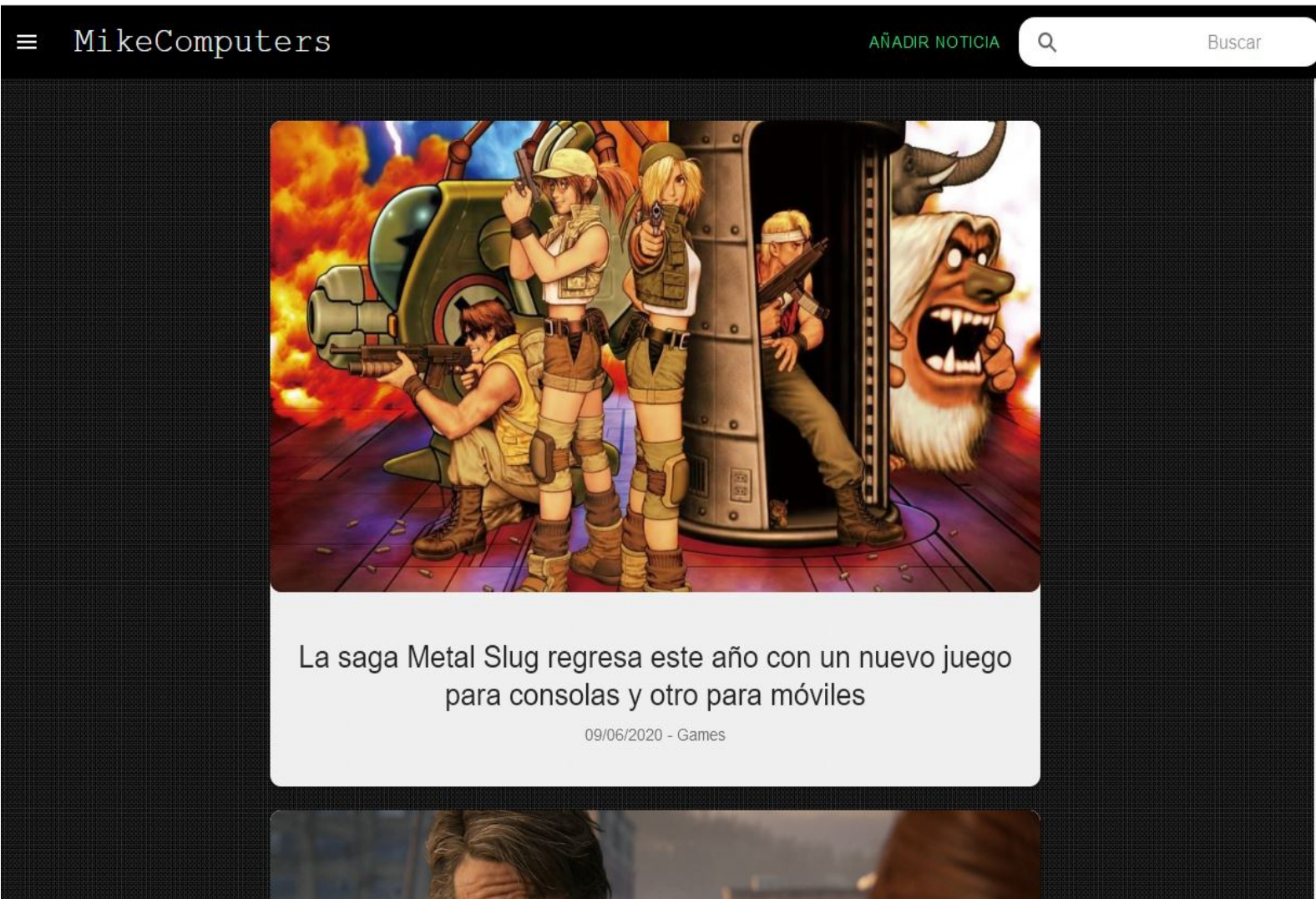


```
1 {
2     "Correcto!": "El cliente ha sido añadido",
3     "usuario": {
4         "nombre": "prueba4",
5         "email": "prueba4@gmail.com",
6         "password": "prueba4",
7         "noticias": [],
8         "id_user": 25
9     }
10 }
```

Parte frontend

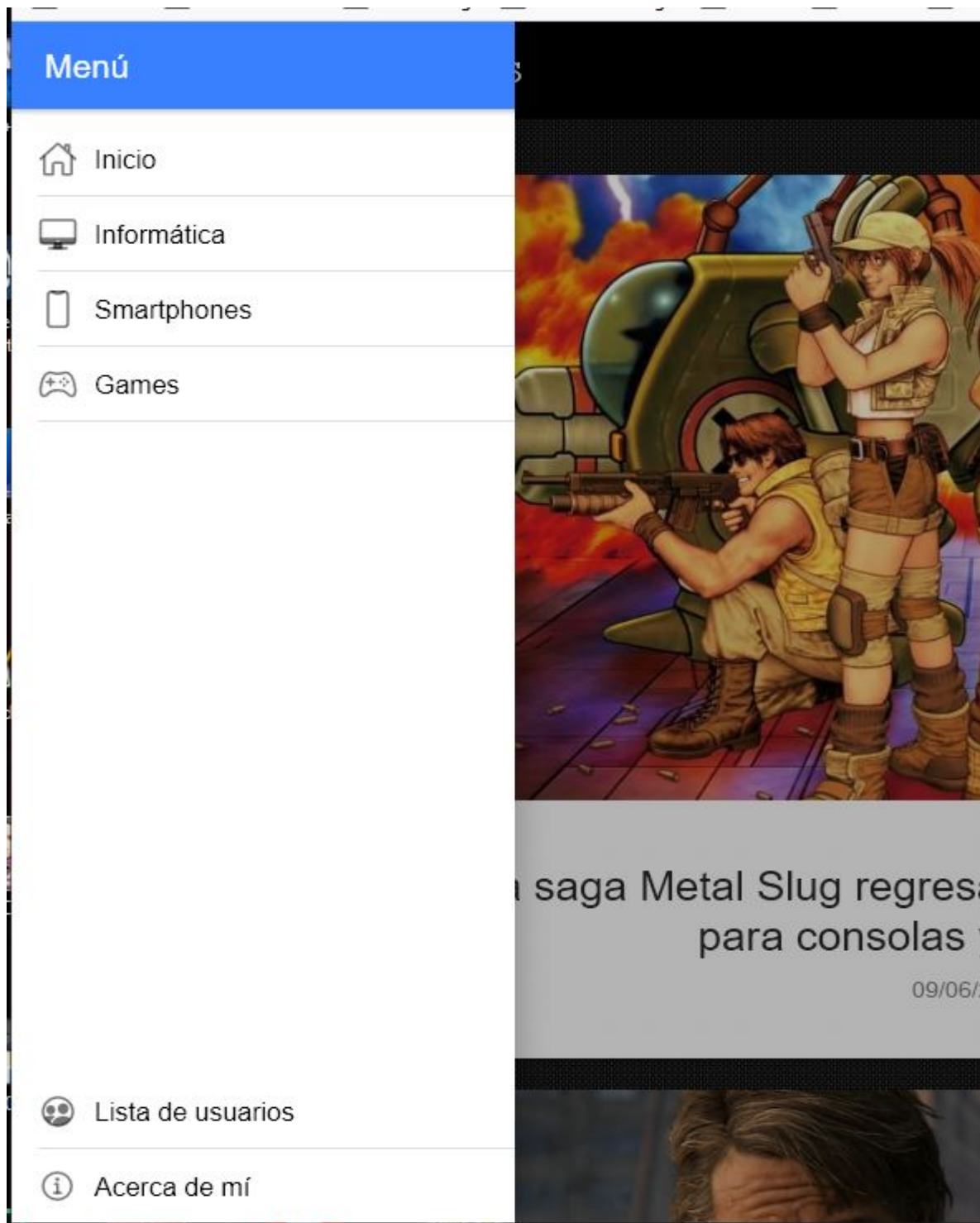
La parte front de la pagina web ha sido desarrollada con Ionic.

Pantalla principal:



En ella podemos encontrar el listado de las noticias que contiene nuestra base de datos. Tenemos las opciones de añadir una noticia nueva, buscar la noticia y ver los detalles de la noticia.

Menu lateral:



Tenemos un menú lateral despegable en el que podemos encontrar las diferentes opciones para filtrar las noticias por su categoría.

También tenemos la opción para acceder a la lista de usuarios y a una sección donde se da información sobre este proyecto y su desarrollador.

Detalle de noticia:

 MikeComputers

09/06/2020 - Games

La Saga Metal Slug Regresa Este Año Con Un Nuevo Juego Para Consolas Y Otro Para Móviles



SNK lleva algún tiempo planeando el regreso de la saga Metal Slug. Ya en abril del año pasado supimos que el desarrollo había comenzado, pero hasta ahora no se ha hablado de

Al hacer “click” sobre una de las noticias de la lista, podremos ver todo su contenido al completo.

En esta vista detallada, también tenemos las opciones de editar o eliminar la noticia.

Añadir noticia:

← MikeComputers

Título *

Categoría: *

Autor: *

Imagen*

Examinar...

No se ha seleccionado ningún archivo.

Contenido*

AÑADIR NOTICIA

CANCELAR

Listado de usuarios:

← MikeComputers

AÑADIR USUARIO



Buscar

EDITAR

Daniel

daniel@gmail.com

Juanma

juanma@gmail.com

Amparo

amparo@gmail.com

Adrian

adrian@gmail.com

Paula

paula@gmail.com

Michael

michael@gmail.com

En la lista de usuarios podemos ver todos los usuarios de la web.
En ella podemos editar, borrar y añadir usuarios.