

# Proyecto

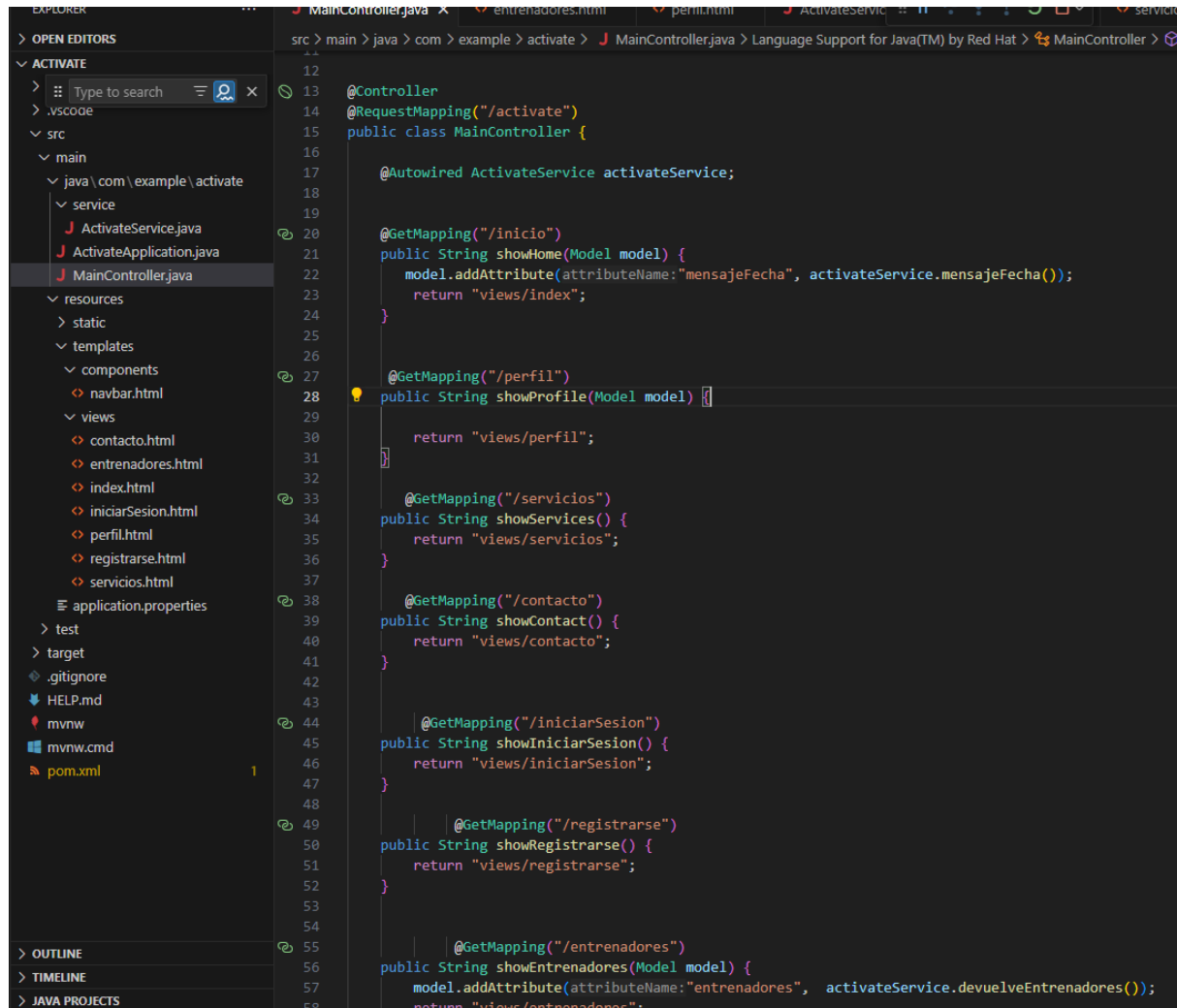
Entrena-te.



<b>Entrena.</b>	<b>1</b>
Pasar todas las páginas estáticas a ser gestionadas mediante un solo controlador llamado MainController.	3
Incluir algún contenido dinámico simple:	3
Añadir fragmentos Thymeleaf que incluyan las partes comunes de todas las vistas como puede ser el head o el menú de navegación.	4
	7

Pasar todas las páginas estáticas a ser gestionadas mediante un solo controlador llamado

MainController.



```
12
13 @Controller
14 @RequestMapping("/activate")
15 public class MainController {
16
17     @Autowired ActivateService activateService;
18
19
20     @GetMapping("/inicio")
21     public String showHome(Model model) {
22         model.addAttribute(attributeName:"mensajeFecha", activateService.mensajeFecha());
23         return "views/index";
24     }
25
26
27     @GetMapping("/perfil")
28     public String showProfile(Model model) {
29
30         return "views/perfil";
31     }
32
33     @GetMapping("/servicios")
34     public String showServices() {
35         return "views/servicios";
36     }
37
38     @GetMapping("/contacto")
39     public String showContact() {
40         return "views/contacto";
41     }
42
43
44     @GetMapping("/iniciarSesion")
45     public String showIniciarSesion() {
46         return "views/iniciarSesion";
47     }
48
49     @GetMapping("/registrarse")
50     public String showRegistrarse() {
51         return "views/registrarse";
52     }
53
54
55     @GetMapping("/entrenadores")
56     public String showEntrenadores(Model model) {
57         model.addAttribute(attributeName:"entrenadores", activateService.devuelveEntrenadores());
58         return "views/entrenadores";
59     }
60 }
```

He creado todos los GetMappings para cada uno de los enlaces principales asociandolos con sus respectivos html.

Incluir algún contenido dinámico simple:

por ejemplo, el año en curso en la página de inicio, listas de elementos (clientes, productos, etc.) con valores literales almacenados en colecciones en el servidor.

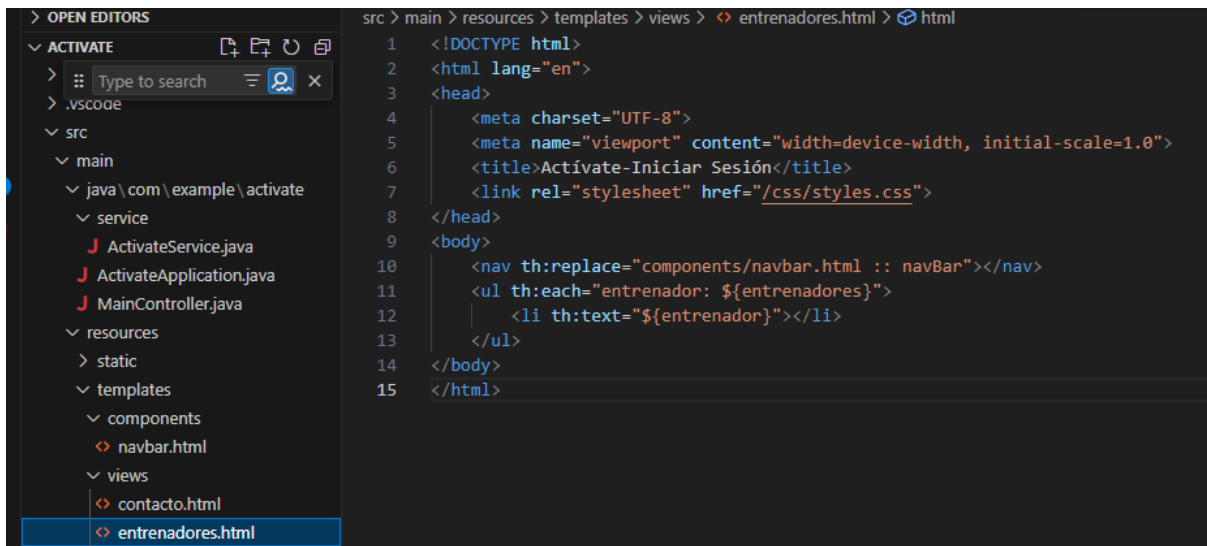
(por ahora estarán en el controlador, aunque no será su ubicación definitiva).

```
src > main > java > com > example > activate > service > ActivateService.java > Language Support for Java(TM) by H
1 import java.util.Calendar;
2 import java.util.ArrayList;
3
4 import java.util.Calendar;
5
6 import org.springframework.stereotype.Service;
7
8 @Service
9 public class ActivateService {
10
11     ArrayList<String> entrenadores = new ArrayList<String>() {
12         {
13             add("Juan González");
14             add("María Rodríguez");
15             add("Pedro Fernández");
16             add("Ana López");
17             add("Luis Martínez");
18             add("Carmen Sánchez");
19             add("Javier Pérez");
20             add("Isabel Gómez");
21             add("Miguel Martín");
22             add("Laura Jiménez");
23         }
24     };
25
26     public String mensajeFecha() {
27
28         Calendar cal = Calendar.getInstance();
29
30         return "Cambia tu " + cal.get(Calendar.YEAR) + "!";
31     }
32
33     public ArrayList<String> devuelveEntrenadores() {
34         return entrenadores;
35     }
36
37 }
38
39
```

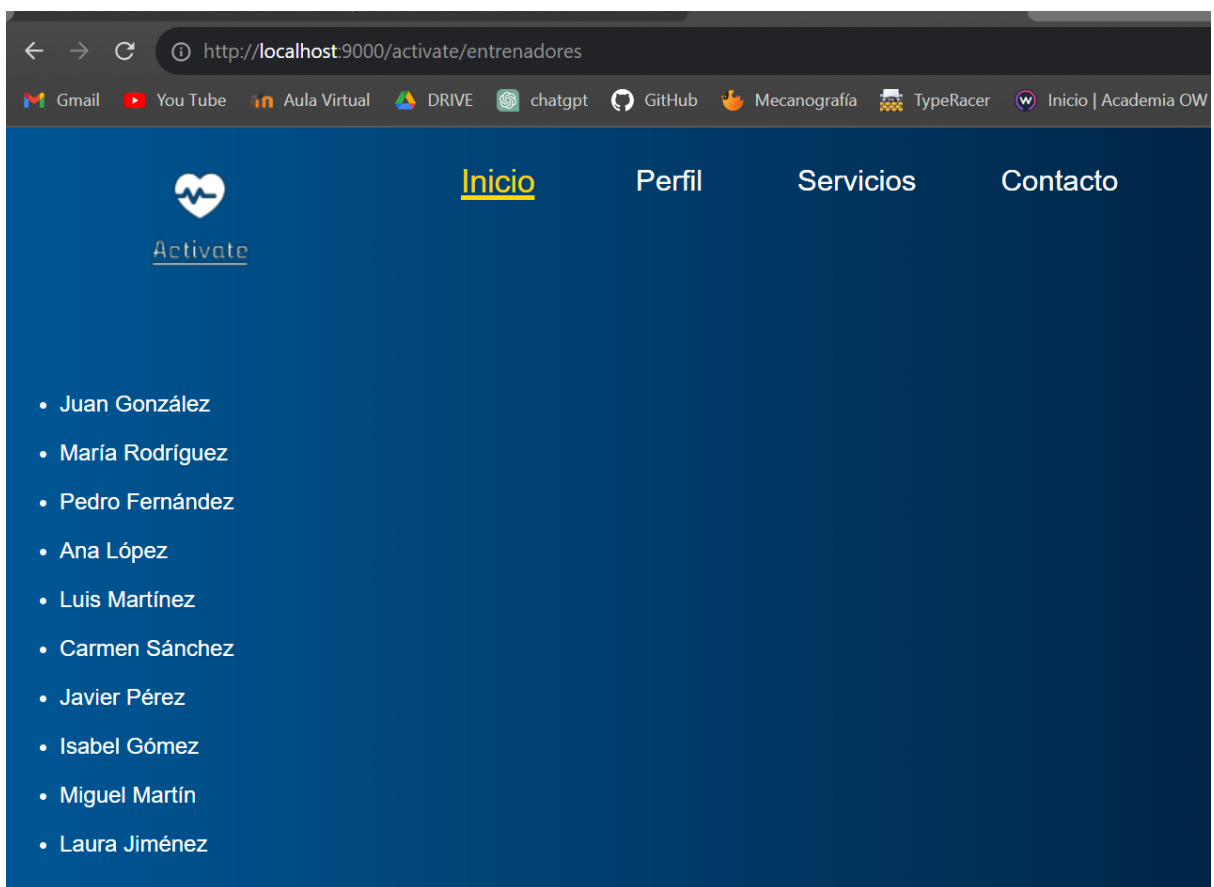
He creado una carpeta “service” para poder separar las funciones de lógica de negocio de la vista y el controlador.

He creado una lista de entrenadores para pasar de forma dinámica al controlador de la página de entrenadores y así mostrar todos los que hay.

```
@GetMapping("/entrenadores")
public String showEntrenadores(Model model) {
    model.addAttribute(attributeName:"entrenadores", activateService.devuelveEntrenadores());
    return "views/entrenadores";
}
```



```
src > main > resources > templates > views > entrenadores.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Actíivate-Iniciar Sesión</title>
7   <link rel="stylesheet" href="/css/styles.css">
8 </head>
9 <body>
10   <nav th:replace="components/navbar.html :: navBar"></nav>
11   <ul th:each="entrenador: ${entrenadores}">
12     <li th:text="${entrenador}"></li>
13   </ul>
14 </body>
15 </html>
```



como podemos ver los muestra correctamente.

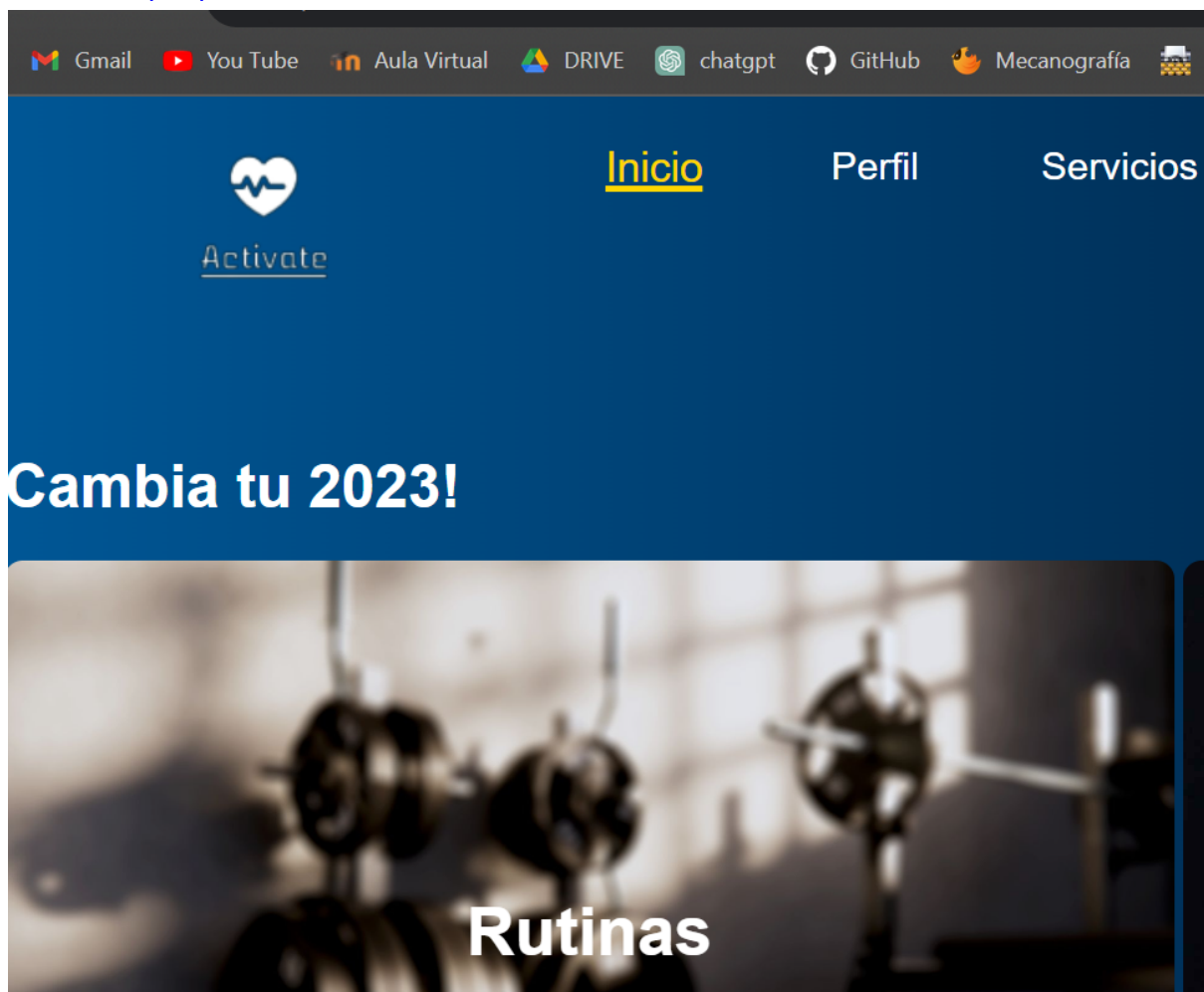
```
@GetMapping("/inicio")
public String showHome(Model model) {
    model.addAttribute(attributeName:"mensajeFecha", activateService.mensajeFecha());
    return "views/index";
}
```

```
</head>

<body cz-shortcut-listen="true">

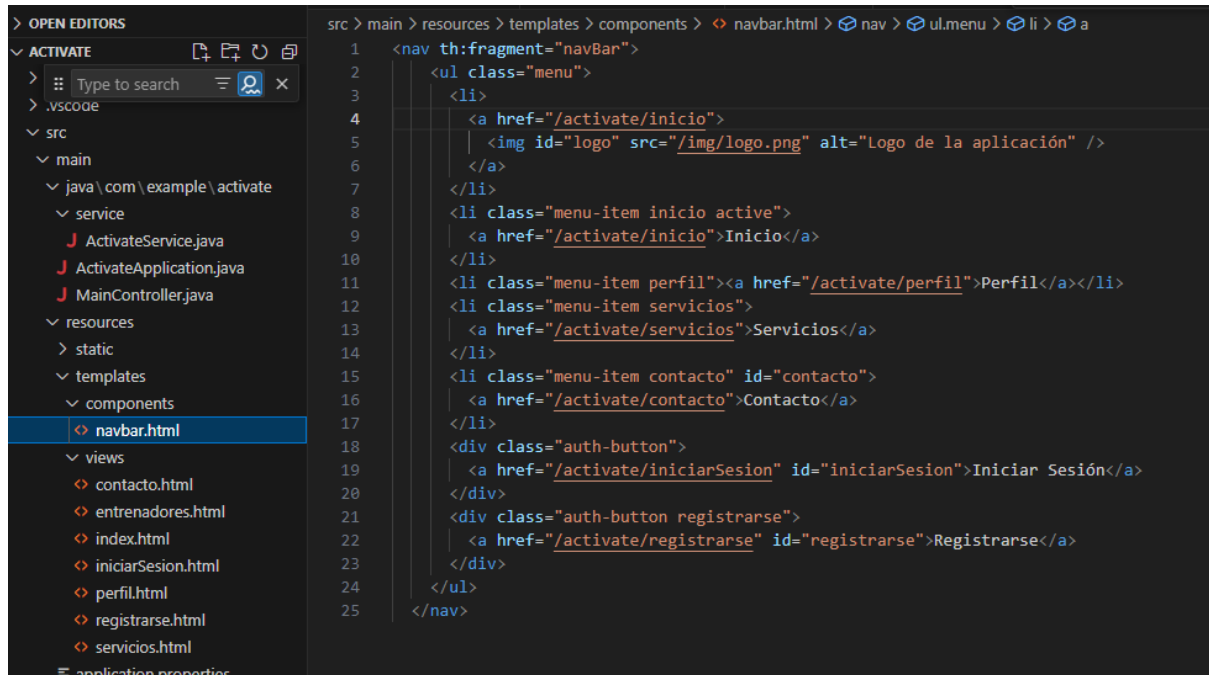
    <nav th:replace="components/navbar.html :: navBar"></nav>
    <h1 th:text = "${mensajeFecha}"></h1>
    <div class="noticias-container">
        <a href="" class="container rutinas">
            <h2>Rutinas</h2>
        </a>
    </div>
</body>
```

Para la página de inicio he llamado a la función que devuelve un mensaje con el año actual y la muestro por pantalla en inicio.



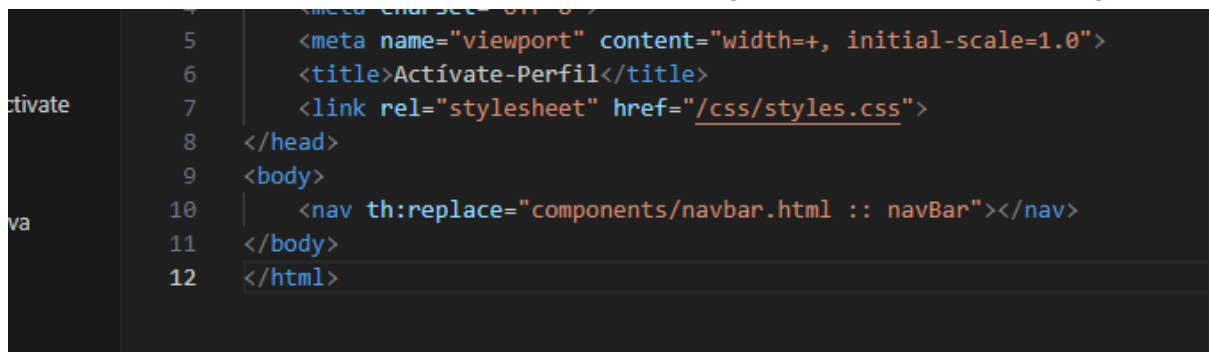
Como podemos ver funciona correctamente.

Añadir fragmentos Thymeleaf que incluyan las partes comunes de todas las vistas como puede ser el head o el menú de navegación.



```
1 <nav th:fragment="navBar">
2   <ul class="menu">
3     <li>
4       <a href="/activate/inicio">
5         
6       </a>
7     </li>
8     <li class="menu-item inicio active">
9       <a href="/activate/inicio">Inicio</a>
10    </li>
11    <li class="menu-item perfil"><a href="/activate/perfil">Perfil</a></li>
12    <li class="menu-item servicios">
13      <a href="/activate/servicios">Servicios</a>
14    </li>
15    <li class="menu-item contacto" id="contacto">
16      <a href="/activate/contacto">Contacto</a>
17    </li>
18    <div class="auth-button">
19      <a href="/activate/iniciarSesion" id="iniciarSesion">Iniciar Sesión</a>
20    </div>
21    <div class="auth-button registrarse">
22      <a href="/activate/registrarse" id="registrarse">Registrarse</a>
23    </div>
24  </ul>
25 </nav>
```

Para dividir mejor los templates he creado una carpeta específica para los componentes. En ella he creado un “navbar” que va a contener el código html de mi barra de navegación.



```
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Actíivate-Perfil</title>
7 <link rel="stylesheet" href="/css/styles.css">
8 </head>
9 <body>
10   <nav th:replace="components/navbar.html :: navBar"></nav>
11 </body>
12 </html>
```

solo tengo que usar “th:replace” para usar mi barra de nav, así no se repite el código y cada vez que quiera cambiar algo me puedo ir directamente a navbar.html y cambiarla.