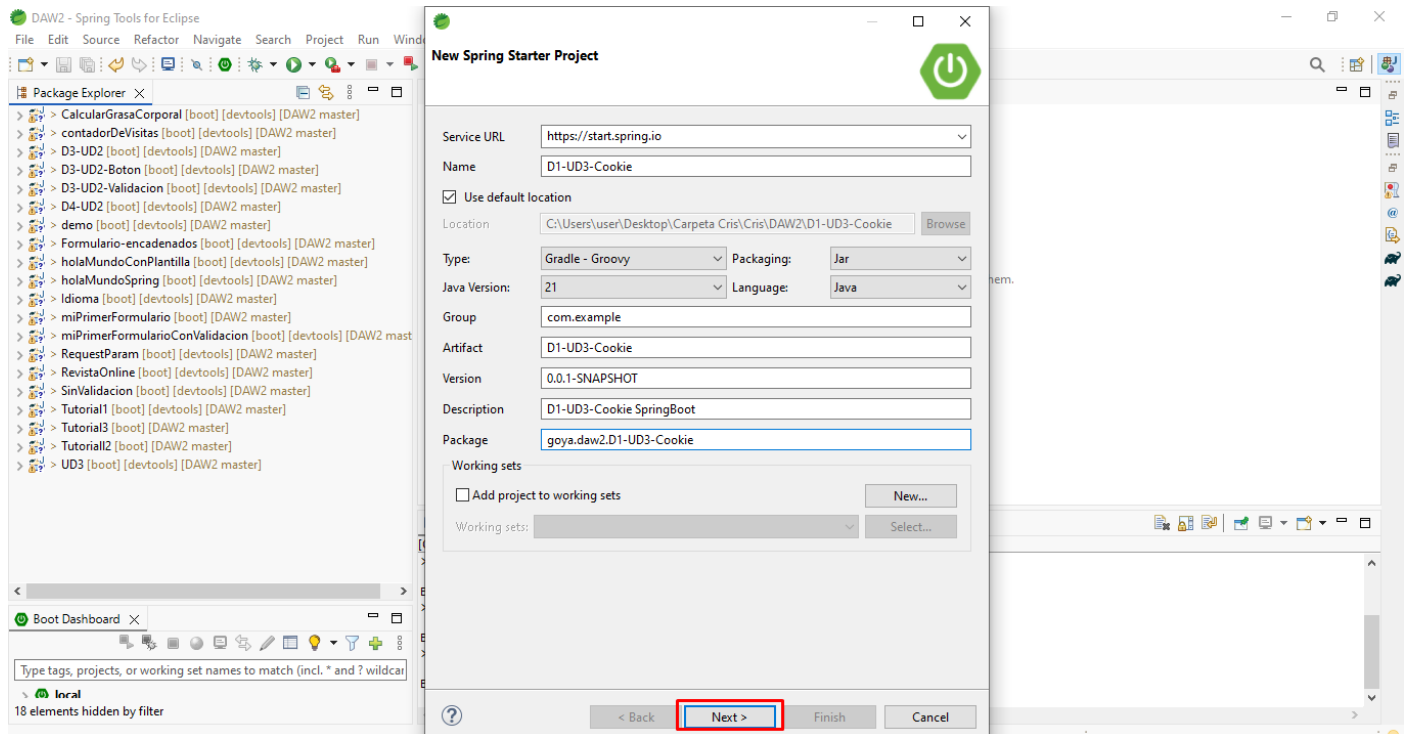


# D1-UD3 enlazando formularios sin campos hidden

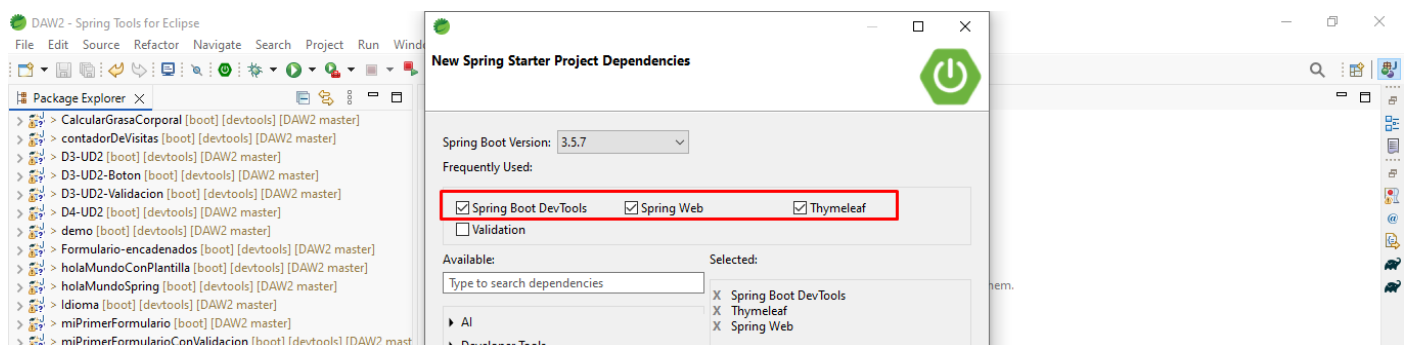
Se trata de modificar alguno de los retos previos o el adjunto para no usar campos ocultos (hidden, salvo para la etapa)

## 1. Con Cookies

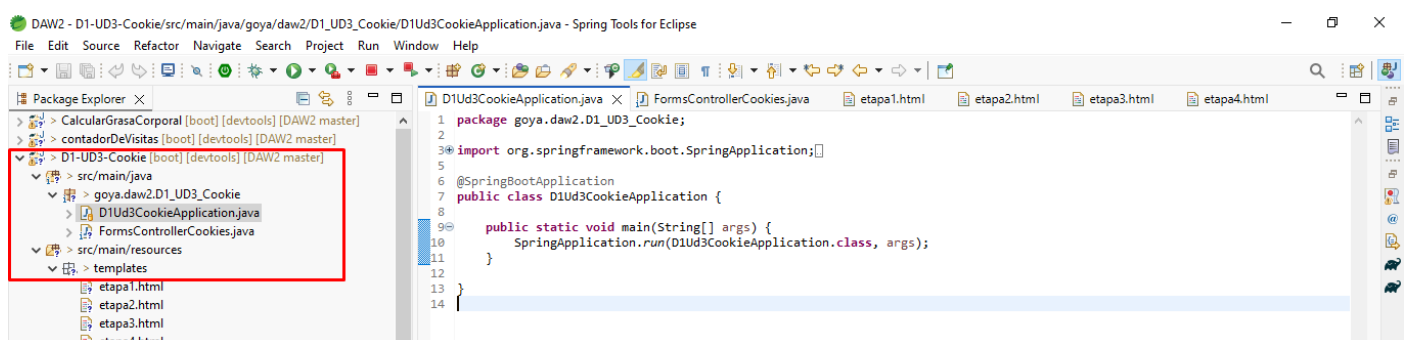
Nos creamos un proyecto llamado D1-UD3-Cookie



Añadimos las dependencias



Vemos que se ha creado el proyecto con la clase por defecto



Creamos la clase `FormsControllerCookies`, gestiona las etapas usando cookies para guardar y recuperar nombre, signo y aficciones, validando los datos en cada etapa. Finalmente muestra el resultado en la etapa 4 sin usar campos ocultos salvo `numEtapa`.

```

1 package goya.daw2.D1_UD3_Cookie;
2
3 import java.util.ArrayList;
4
5 @Controller
6 public class FormsControllerCookies {
7
8     static final String[] SIGNOS = { "", "Aries", "Tauro", "Géminis", "Cáncer", "Leo", "Virgo", "Libra", "Escorpio",
9         "Sagitario", "Capricornio", "Acuario", "Piscis" };
10
11     static final String[] AFICCIONES = { "Deportes", "Juerga", "Lectura", "Relaciones sociales" };
12
13     @GetMapping("/cookies")
14     String getetapa1(@CookieValue(value = "nombre", required = false) String nombreCookie, Model model) {
15         model.addAttribute("numEtapa", 1);
16         if (nombreCookie != null) {
17             model.addAttribute("nombre", nombreCookie);
18         }
19         return "etapa1";
20     }
21
22     @PostMapping("/cookies")
23     String procesaetapa(@RequestParam(name = "numEtapa") Integer numEtapa,
24         @RequestParam(name = "nombre", required = false) String nombre,
25         @RequestParam(name = "signo", required = false) String signo,
26         @RequestParam(name = "aficciones", required = false) String[] aficciones,
27         @CookieValue(value = "nombre", required = false) String nombreCookie,
28         @CookieValue(value = "signo", required = false) String signoCookie,
29         @CookieValue(value = "aficciones", required = false) String aficcionesCookie, HttpServletResponse response,
30         Model model) {
31
32         modelo.addAttribute("signos", SIGNOS);
33         modelo.addAttribute("aficciones", AFICCIONES);
34
35         // Nombre
36         if (nombre == null || nombre.isBlank()) {
37             nombre = nombreCookie;
38         } else {
39             Cookie cookieNombre = new Cookie("nombre", nombre);
40             cookieNombre.setMaxAge(3600);
41             response.addCookie(cookieNombre);
42         }
43
44         // Signo
45         if (signo == null) {
46             signo = signoCookie;
47         } else {
48             Cookie cookieSigno = new Cookie("signo", signo);
49             cookieSigno.setMaxAge(3600);
50             cookieSigno.setHttpOnly(true);
51             cookieSigno.setPath("/cookies");
52             response.addCookie(cookieSigno);
53         }
54
55         // Aficciones
56         if (aficciones == null) {
57             if (aficcionesCookie != null) {
58                 aficciones = aficcionesCookie.split(",");
59             }
60         } else {
61             Cookie cookieAficciones = new Cookie("aficciones", String.join(",", aficciones));
62             cookieAficciones.setMaxAge(3600);
63             cookieAficciones.setHttpOnly(true);
64             cookieAficciones.setPath("/cookies");
65             response.addCookie(cookieAficciones);
66         }
67
68         modelo.addAttribute("nombre", nombre);
69         modelo.addAttribute("signo", signo);
70         modelo.addAttribute("numEtapa", numEtapa);
71
72         // Validaciones
73     }
74 }

```

```

1 package goya.daw2.D1_UD3_Cookie;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class D1UD3CookieApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(D1UD3CookieApplication.class, args);
11     }
12 }

```

## Validaciones

```

80      modelo.addAttribute("nombre", nombre);
81      modelo.addAttribute("signo", signo);
82      modelo.addAttribute("numEtapa", numEtapa);
83
84      // Validaciones
85      String errores = "";
86      if (numEtapa == 1 && (nombre == null || nombre.isBlank())) {
87          errores = "Debes poner un nombre no vacío";
88      } else if (numEtapa == 1 && (nombre.length() < 3 || nombre.length() > 10)) {
89          errores = "La longitud del nombre debe estar entre 3 y 10";
90      }
91      if (numEtapa == 2 && (signo == null || signo.equals(""))) {
92          errores = "Debes seleccionar un signo";
93      }
94      if (numEtapa == 3 && (aficciones == null || aficciones.length == 0)) {
95          errores = "Debes elegir al menos una afición, no seas soso/a";
96      }
97
98      if (!errores.isBlank()) {
99          modelo.addAttribute("errores", errores);
100         return "etapa" + numEtapa;
101     }
102
103     numEtapa++;
104     modelo.addAttribute("numEtapa", numEtapa);
105
106     if (numEtapa == 4) {
107         List<String> respuestas = new ArrayList<>();
108         respuestas.add(nombre);
109         respuestas.add(signo);
110         if (aficciones == null) {
111             aficciones = new String[0];
112         }
113         respuestas.add(String.join(", ", aficciones));
114         modelo.addAttribute("respuestas", respuestas);
115     }
116     return "etapa" + numEtapa;
117 }
118
119

```

**Creamos un HTML llamado etapa1.html, donde se pide al usuario que introduzca su nombre. Muestra errores si los hay y envía el formulario con numEtapa=1 para avanzar a la siguiente etapa**

```

1 <!DOCTYPE HTML>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head><title>MiniQuiz etapa 1</title></head>
4 <body>
5 <form method="post" th:action="@{/cookies}">
6     <p th:text="${errores}"></p>
7     <label for="nombre">Nombre:</label>
8     <input type="text" name="nombre" th:value="${nombre}" />
9     <input type="hidden" name="numEtapa" value="1" />
10    <input type="submit" value="Siguiente" />
11 </form>
12 </body>
13 </html>
14

```

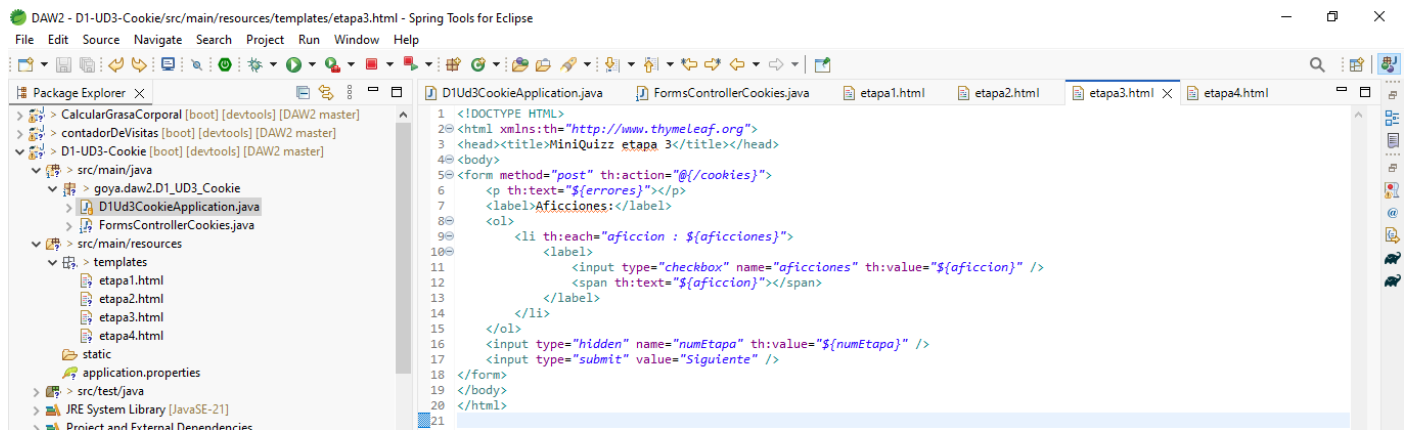
**Creamos un HTML llamado etapa2.html, donde el usuario selecciona su signo del zodiaco. Muestra errores si los hay y envía el formulario con numEtapa=2 para pasar a la etapa siguiente**

```

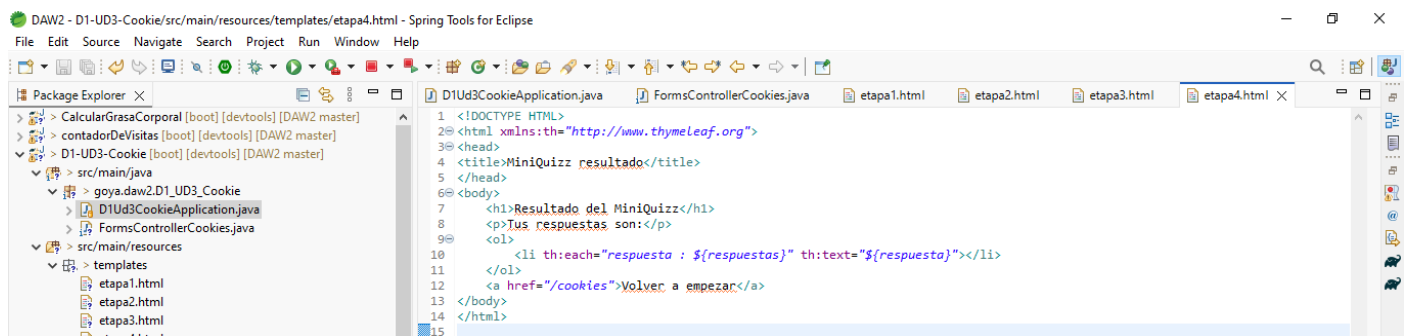
1 <!DOCTYPE HTML>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head><title>MiniQuiz etapa 2</title></head>
4 <body>
5 <form method="post" th:action="@{/cookies}">
6     <p th:text="${errores}"></p>
7     <label for="signo">Signo:</label>
8     <select name="signo">
9         <option th:each="signo, indice : ${signos}"
10             th:value="${signo.index}"
11             th:text="${signo}"></option>
12     </select>
13     <input type="hidden" name="numEtapa" th:value="${numEtapa}" />
14     <input type="submit" value="Siguiente" />
15 </form>
16 </body>
17 </html>
18

```

Creamos un HTML llamado etapa3.html, donde el usuario selecciona sus aficciones. Muestra errores si los hay y envía el formulario con numEtapa=3 para pasar a la etapa de resultados.



Creamos un HTML llamado etapa4.html, donde se muestran todas las respuestas en una lista ordenada, con un enlace para volver a empezar.



Navegador: <http://localhost:8080/cookies>

Lo primero que nos pide es el nombre

El signo y tenemos que elegir uno

## Aficiones, es obligatorio poner uno

localhost:8080/cookies

Debes elegir al menos una aficción, no seas soso/a

Aficciones:

- ☐ Deportes
- ☐ Juerga
- ☐ Lectura
- ☐ Relaciones sociales

## Resultado

localhost:8080/cookies

## Resultado del MiniQuizz

Tus respuestas son:

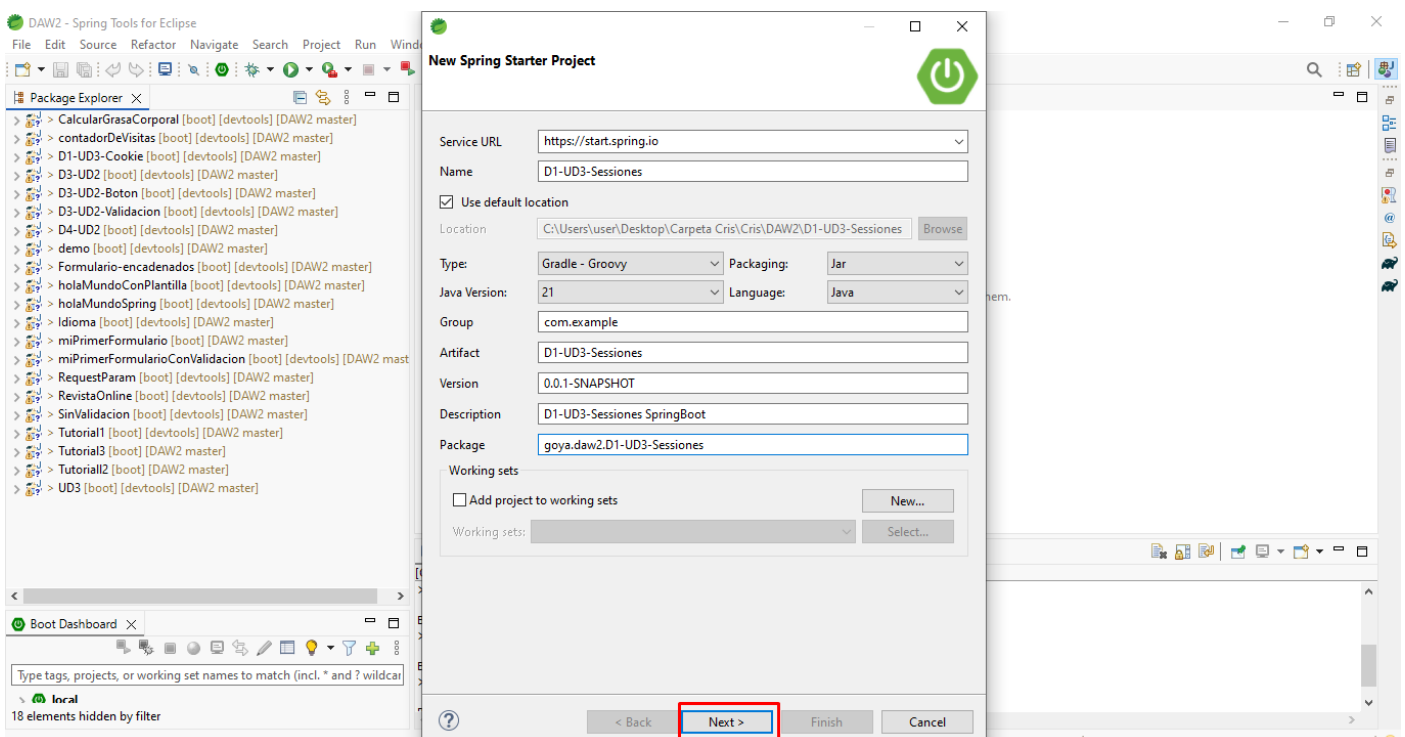
1. Cris
2. Virgo
3. Deportes

[Volver a empezar](#)

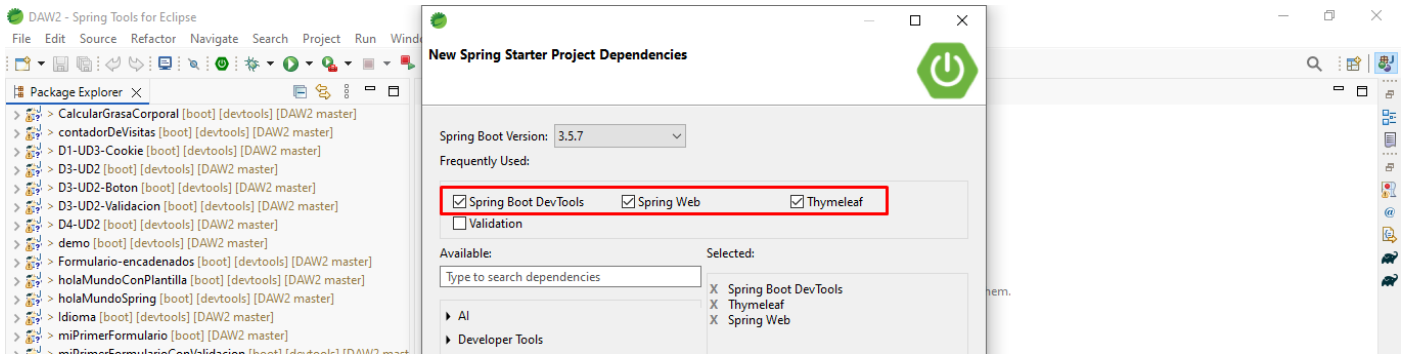
## 2. Con Sesiones

Nota: hay que hacerlo con RequestParams pues todavia no hemos visto como ligar un ModelAttribute con Cookie o Session. Si hiciste el reto previo con ModelAttribute (objeto) te sugiero usar el proyecto adjunto (importar como proyecto Gradle) o hacer uno nuevo desde cero.

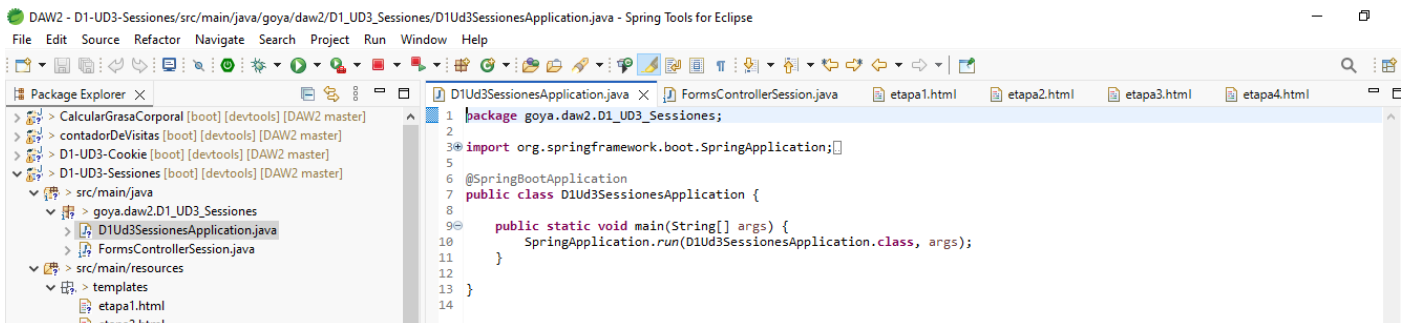
## Nos creamos un proyecto llamado D1-UD3-Sesiones



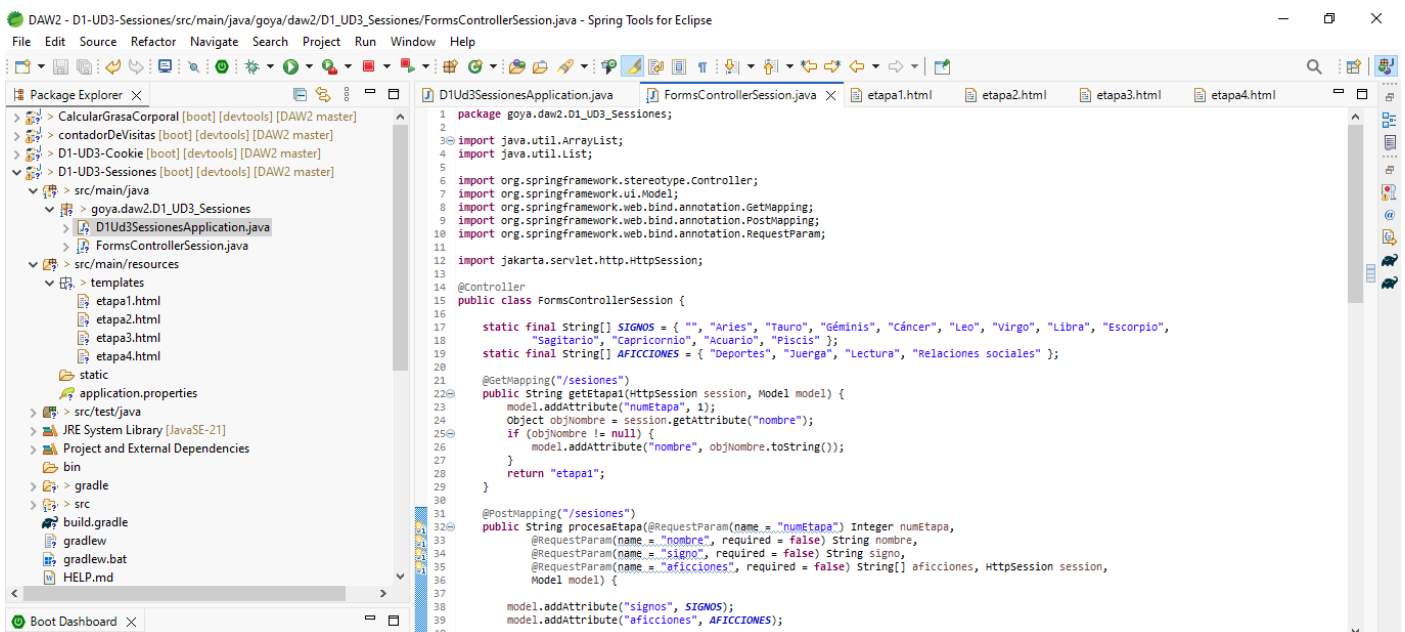
## Añadimos las dependencias



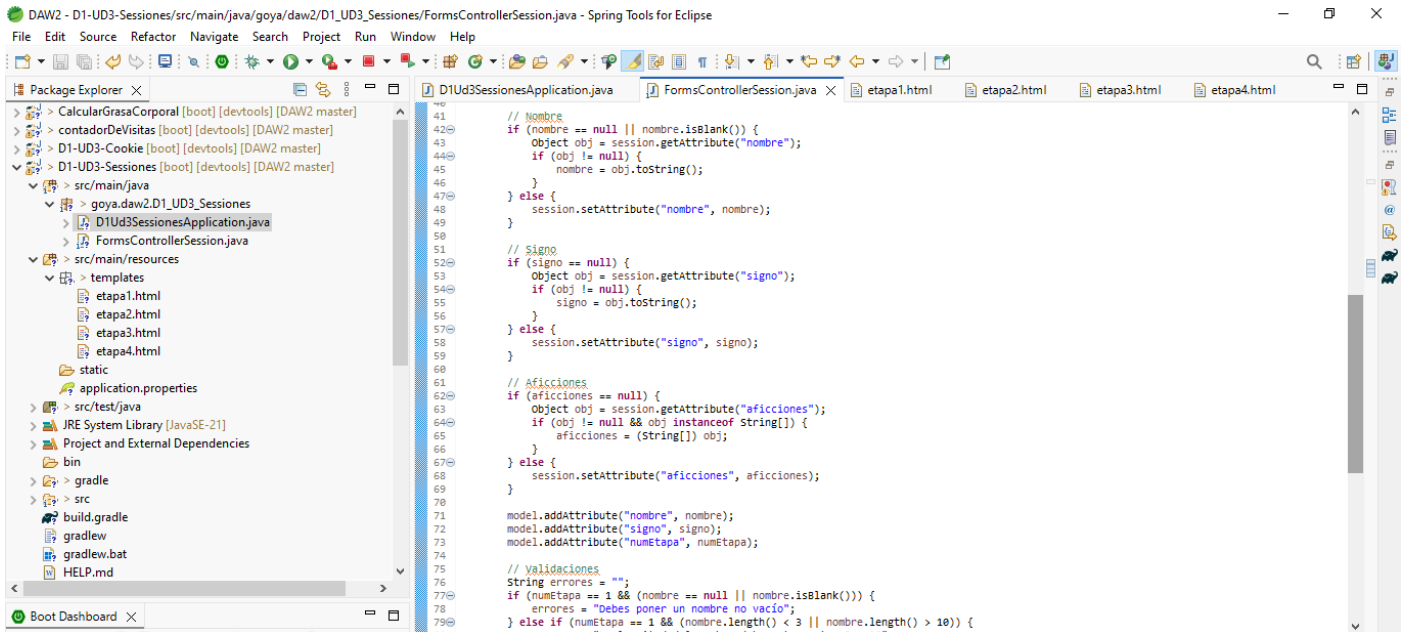
## Vemos que se ha creado el proyecto y la clase por defecto



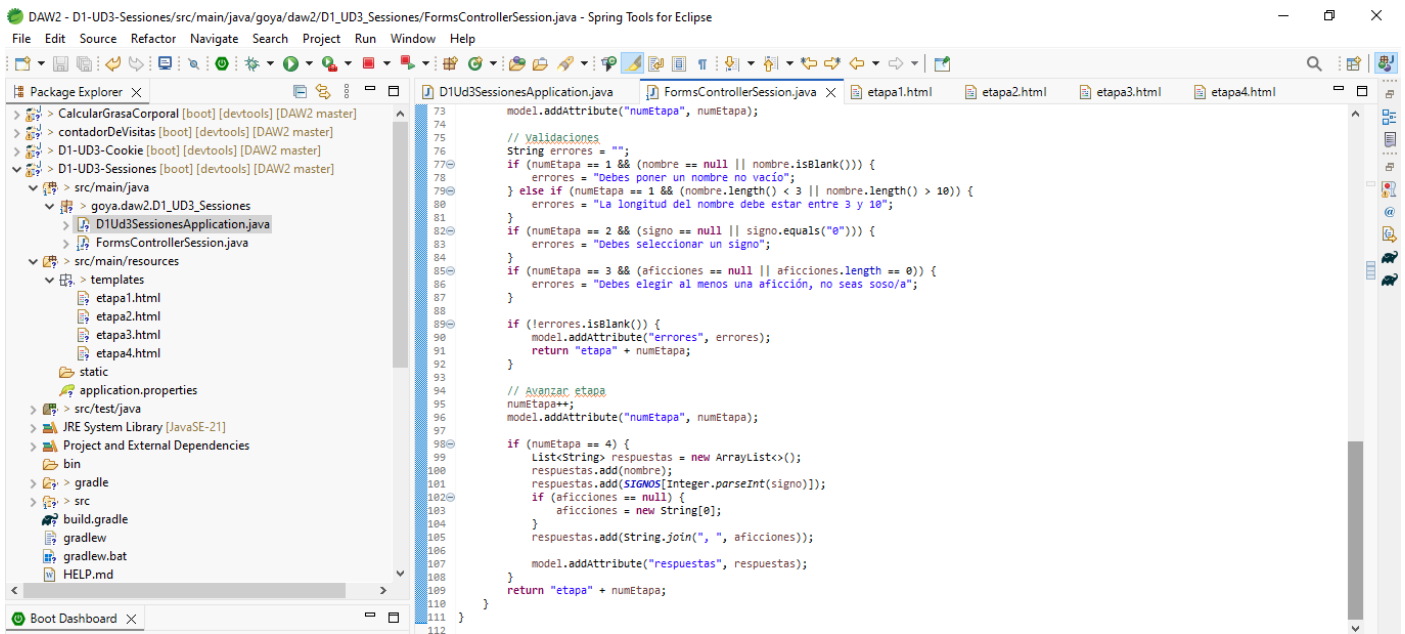
## Creamos la clase FormsControllerSession



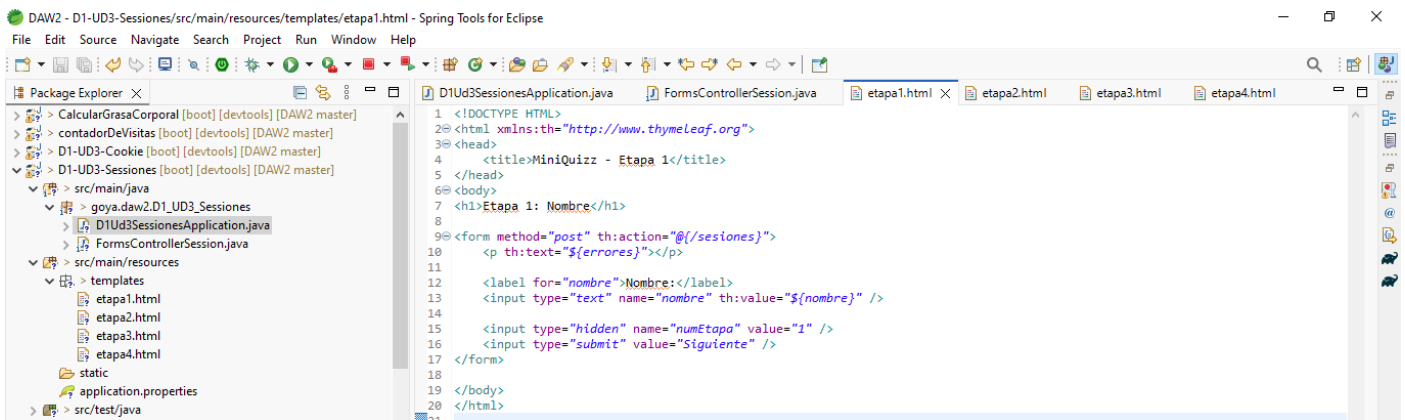




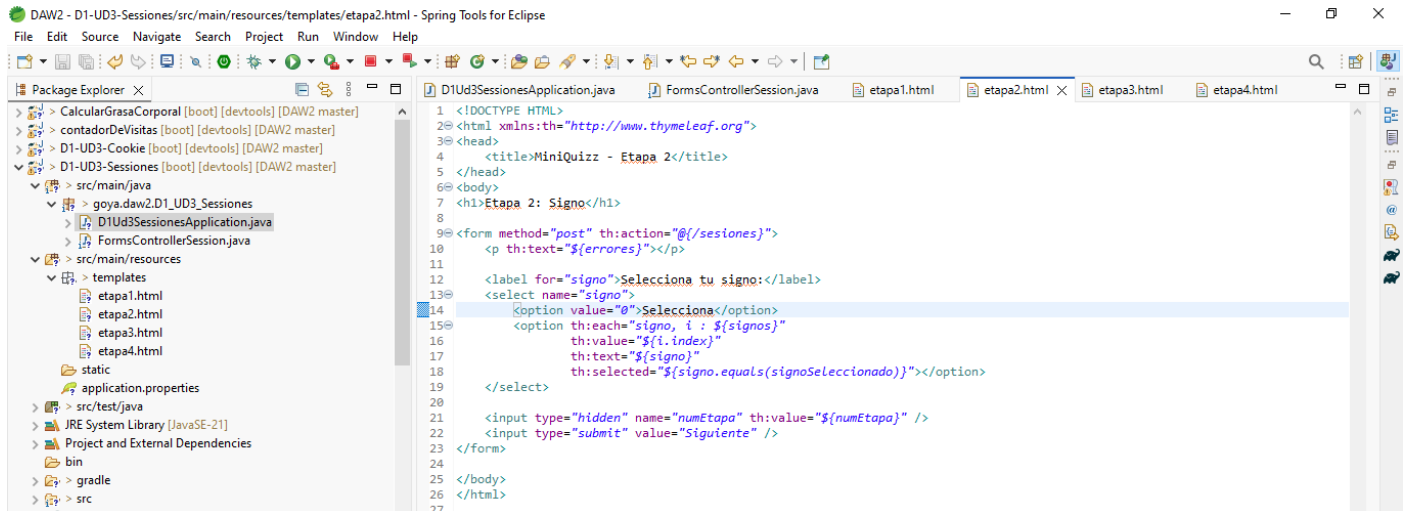
## Validaciones



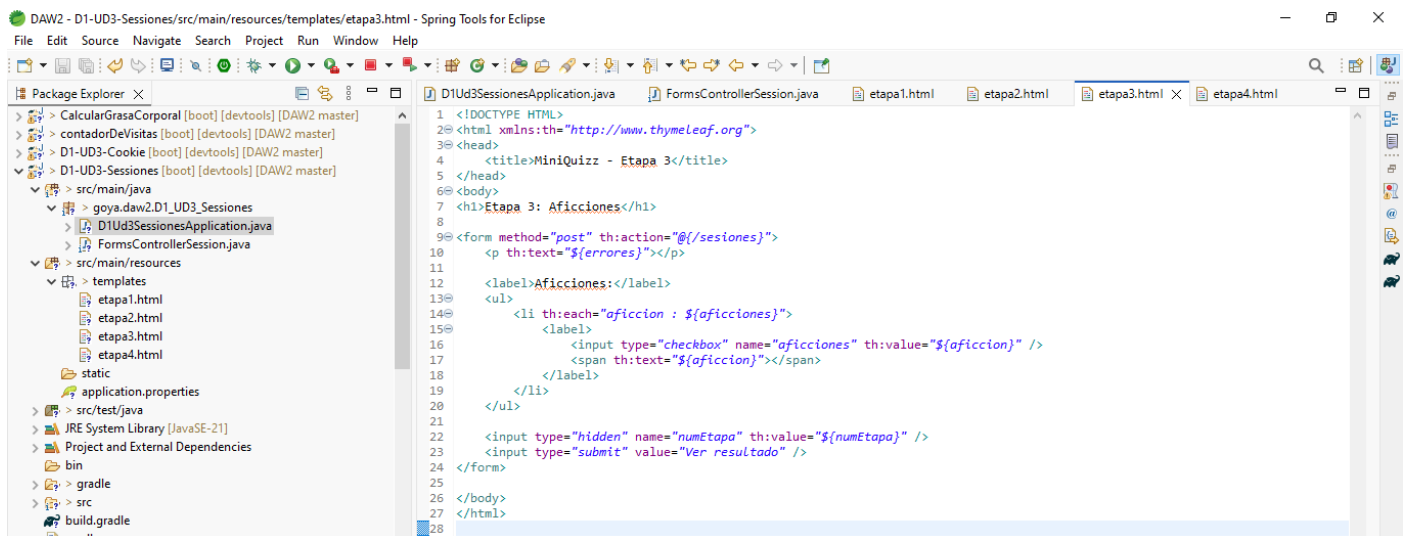
## Creamos un HTML llamado etapa1.html



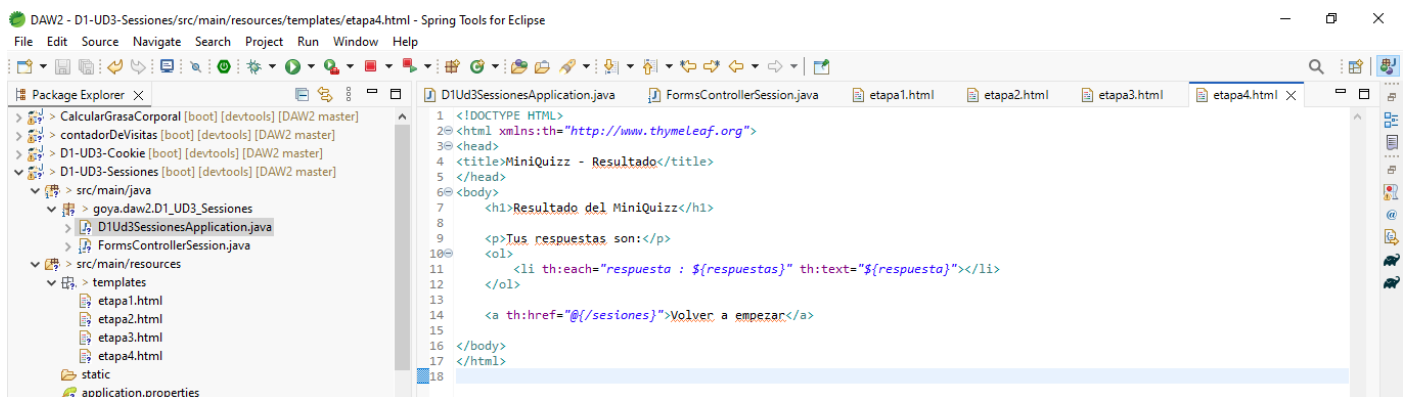
## Creamos un HTML llamado etapa2.html



## Creamos un HTML llamado etapa3.html



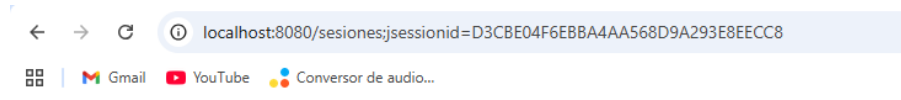
## Creamos un HTML llamado etapa4.html





Navegador: http://localhost:8080/sesiones

## Ponemos nuestro nombre

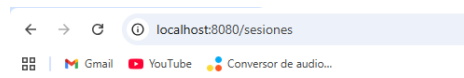


### Etapa 1: Nombre

Debes poner un nombre no vacío

Nombre:

## El signo



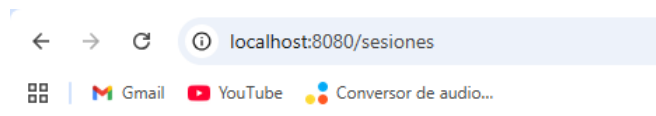
### Etapa 2: Signo

Debes seleccionar un signo

Selecciona tu signo:

- Selecciona
- Aries
- Tauro
- Géminis
- Cáncer
- Leo
- Virgo
- Libra
- Escorpio
- Sagitario
- Capricornio
- Acuario
- Piscis

## Aficiones



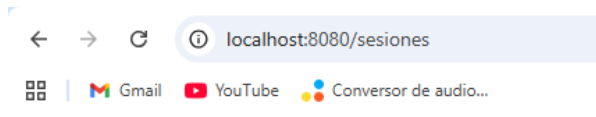
### Etapa 3: Aficciones

Debes elegir al menos una aficción, no seas soso/a

Aficciones:

- ☐ Deportes
- ☐ Juerga
- ☐ Lectura
- ☐ Relaciones sociales

## Resultado



## Resultado del MiniQuizz

Tus respuestas son:

1. Cris
2. Cáncer
3. Juega, Lectura

[Volver a empezar](#)

Entrega: documento D1-UD3.PDF con la descripción de los pasos realizados, problemas encontrados, soluciones aportadas y conclusiones sobre la consecución de objetivos en cada apartado. El código se entregará en un enlace a cloud o git, ojo al .gitignore.

Evaluación:

- Funcionalidad correcta.
- Documentación clara y representativa del trabajo realizado.

Calificación:

- Suficiente: uno de los apartados funcional y documentado.
- Bien: los dos apartados funcionales y documentados.
- Excelente: lo anterior pero con una documentación notable que incluya una reflexión sobre qué cambia en el ciclo de petición/respuesta y dónde se guardan los datos entre el apartado 1 y el 2 (no hace falta mencionar los cambios en el código)

**Con cookies, los datos del usuario se guardan en el navegador del cliente y se envían al servidor con cada nueva petición HTTP.**

**Con sesiones los datos se almacenan en el servidor y el navegador solo guarda un identificador de sesión, el servidor puede recuperar la información del usuario sin reenviarla en cada petición.**

## Enlace GitHub:

**<https://github.com/Cristina200317/PracticasEntornoServidor/tree/8f090451d5b5651df1451b9c5c587fe27876996e/Tema3/D1-UD3>**