

## Práctica 3: SpringQUIZ

En esta práctica el alumno deberá desarrollar un típico Quiz que se encuentra en las redes sociales. En el que a través de una serie de preguntas se determine el grado de experto en dicha materia.

Requisitos para el quiz:

### 1º Cuatro clasificaciones:

El resultado del juego debe dar una clasificación entre al menos 4 disponibles. Cada pregunta supondrá una puntuación añadida para cada marcador de la clasificación. El resultado final será la clasificación con mayores puntos en el marcador.

Sobre este apartado he creado un método que recoja todos los resultados dados en las diferentes respuestas (los míos so numéricos) y se hace una suma de ellos.

Después se mete en un condicional para comprobar a cuál opción final se envía al usuario mediante el resultado de esa suma.



Captura del resultado y la pagina correspondiente al mismo

```

@PostMapping("/resultado")
public String resultado(String q7, HttpSession session) {
    return calcularResultado(q7, session);
}

public String calcularResultado(String q7, HttpSession session) {
    int resultado = 0;
    session.setAttribute("q7", q7);

    for (int i = 1; i < 8; i++) {
        if (session.getAttribute("q" + i) != null)
            resultado += Integer.parseInt(session.getAttribute("q" + i).toString());
    }
    System.out.println(resultado);
    if (resultado > 15) {
        return "Grifindor";
    } else if (resultado > 10) {
        return "Hufflepuf";
    } else if (resultado > 5) {
        return "Ravenclaw";
    } else
        return "Slitherin";
}

```

Captura del método de ver los resultados

## 2º Uso de la SESSION:

Al usuario se le plantarán diversas preguntas en las que se

deberá almacenar la puntuación obtenida. Para almacenar el resultado de las

preguntas entre cada página se verá hacer uso de las sesiones. Se recomienda

simplificar la aplicación y el conteo de estas puntuaciones para que el grueso de las

horas invertidas no recaiga en la lógica de negocio.

El uso de sesiones ha servido para usarlo de almacén de los diferentes resultados.

Directamente uso el método @Post mapping para poder guardar el dato enviado y el setAttribute para guardar el dato recibido del post durante la sesión.



```

@PostMapping("/pregunta7")
public String p7(String q6, HttpSession session) {
    session.setAttribute("q6", q6);
    return "pregunta7";
}

```

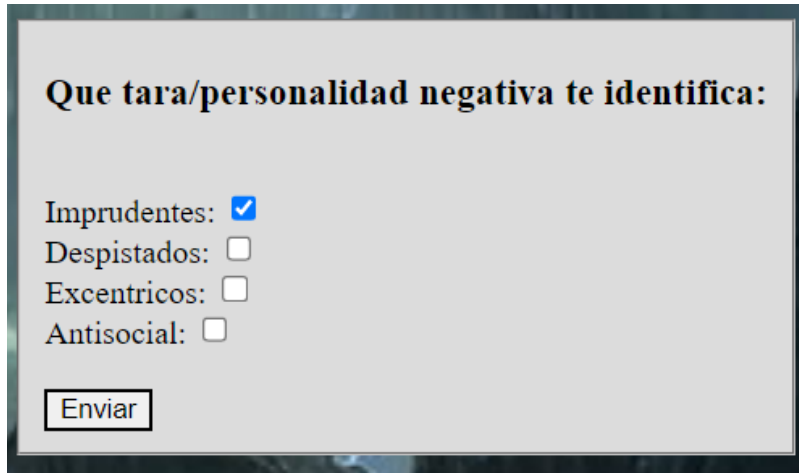
Captura del método de session

### 3ª preguntas:

Se deberá hacer como mínimo 7 preguntas al usuario empleando cada

una de las diferentes entradas al menos una vez: textArea, CheckBox, RadioButton, Button.

Se han creado 7 preguntas al usuario con diferentes entradas para que sea más ameno algunas de ellas repetidas por simplificar.



**Que tara/personalidad negativa te identifica:**

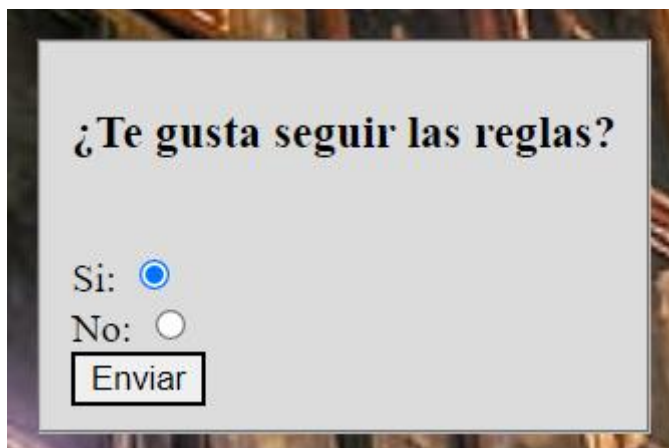
Imprudentes: ☒

Despistados: ☐

Excentricos: ☐

Antisocial: ☐

Captura del Checkbox

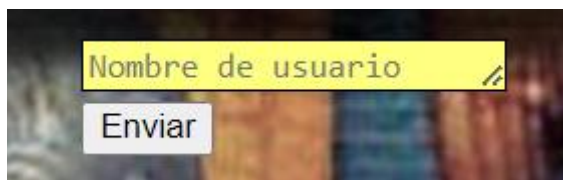


**¿Te gusta seguir las reglas?**

Si: ☒

No: ☐

Captura del radiobutton



Nombre de usuario

Captura del text área y button

## 4º Resultado final:

Habiendo calculado el resultado, para emitir al veredicto se le deberá establecer una categoría al usuario empleando una imagen para dicha categoría.

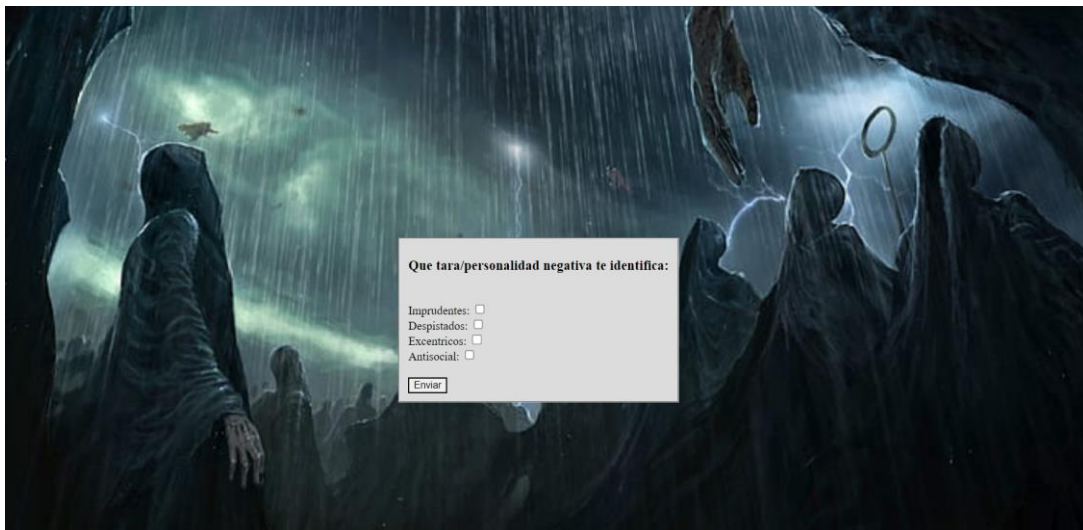
Como remarque en el apartado 1 hay una foto y descripción cuando el usuario termina las respuestas, es el resultado final del test.

(Explicación en el apartado 1 ya mencionada)

## 5º Hojas de estilo:

No se entiende una página desarrollada en HTML sin su CSS, por ello deberás incorporar estilos a los dos anteriores ejercicios.

La Web está diseñada únicamente con css en cada uno de los html.



Captura del diseño de la pagina

```
<style>
  body {
    display: flex;
    justify-content: center;
    background-image: url("img/tara.jpg");
    background-size: cover;
  }

  fieldset {
    margin-top: 80%;
    color: black;
    background-color: gainsboro;
  }

  .button {
    color: black;
    border-color: black;
  }
</style>
```

Captura del código de la página.