

Pregunta 1

Finalizado

Puntuación 0,05 sobre 0,10

🚩 Marcar pregunta

¿A qué elementos afecta una regla CSS cuyo selector es `div[class~="celda"] strong`?

Afecta al elemento `<div>` si su atributo `class` es una lista de valores separados y uno es "celda" y además debe cumplir que esté presente el elemento ``.

Pregunta 2

Sin contestar

Valor: 0,10

🚩 Desmarcar

En el contexto de los formularios de HTML, vimos en clase que el campo de entrada de tipo `hidden` se usaba, en particular, para resolver uno de estos problemas (marque cuál):

- ☐ a. CQRS (*Command and Query Responsibility Segregation*)
- ☐ b. CORS (*Cross-Origin Resource Sharing*)
- ☐ c. CSRF (*Cross-Site Request Forgery*)

Pregunta 3

Finalizado

Puntuación 0,10 sobre 0,10

🚩 Desmarcar

Imagine que usted es ChatGPT y le piden que escriba una frase sobre el brexit en HTML haciendo un marcado correcto de los extranjerismos y los párrafos. ¿Qué escribiría?

```
<p>
    El <i> brexit </i> es importante en momentos críticos
</p>
```

Pregunta 4

Finalizado

Puntuación 0,10 sobre 0,10

🚩 Marcar pregunta

Sea el siguiente código HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Principio de Hamilton</title>
</head>
<body>
  <div class="bloque">
    <div class="hamilton">El <em>principio de Hamilton</em>
    afirma que la evolución de un sistema físico ocurre de tal
    forma que se minimiza la acción del sistema.
    </div>
  </div>
</body>
</html>
```

Al que se le aplica las siguientes reglas de CSS:

```
* {color: red}
.bloque {color: blue}
.hamilton {color: green}
```

Al renderizarlo en el navegador el texto "principio de Hamilton" se ve rojo. Explique por qué se ve de ese color y no azul o verde.

La regla `*` se le aplica a todo elemento por lo que todo pasa a ser rojo.

La regla `.bloque` y `.hamilton` no le afecta porque "principio de Hamilton" está contenido en el elemento ``.

Por eso es rojo y no azul ni verde.

Pregunta 5

Correcta

Puntuación 0,10 sobre 0,10

🚩 Marcar pregunta

¿Qué decorador usa la clase asociada a un componente de Angular para definir atributos de entrada que puede establecer el componente padre para pasarle información?

Respuesta: ✓

Pregunta 6

Finalizado

Puntuación 0,00 sobre 0,10

🚩 Desmarcar

Estaba usando la versión 3.2 de una aplicación Web y ahora quiero usar la nueva versión 3.3, que ya está disponible, ¿qué tengo que instalar en mi máquina? Justifique la respuesta para puntuar.

En la línea de petición de HTTP debo poner la versión HTTP utilizada

Pregunta 7

Sin contestar

Valor: 0,10

🚩 Marcar pregunta

Estamos desarrollando un componente en Angular y hemos incluido el código de plantilla en el fichero `gestion-usuarios.component.html` y las hojas de estilo del componente en `gestion-usuarios.component.css`. Ambos ficheros se encuentran en el mismo directorio que el fichero `.ts` con el código de mi componente. Queremos que nuestro componente sea insertado en otros allá donde se encuentren las etiquetas `<gestion-usuarios>`. Escriba el decorador (y sus argumentos) que es necesario ubicar encima de la clase de mi componente para configurarlo de la forma deseada.

Pregunta 8

Finalizado

Puntuación 0.00 sobre 0.10

🚩 Marcar pregunta

En una clase entidad se encuentra con el siguiente atributo:

```
Set<Alumno> alumnos;
```

La clase Alumno es una clase entidad. Debe anotar ese atributo con exactamente una anotación de JPA, ¿entre cuáles tendría que elegir para que la clase entidad sea correcta?

Para modelar un atributo que es una colección de valores se hace uso de la anotación @ElementCollection.

Para que la clase entidad sea correcta debería usar:

```
@ElementCollection(fetch=FetchType.LAZY)
```

Pregunta 9

Finalizado

Puntuación 0.10 sobre 0.10

🚩 Marcar pregunta

Tenemos una jerarquía de clases entidad con más de una clase donde la raíz de la jerarquía está anotada con @Inheritance(strategy=InheritanceType.SINGLE_TABLE). Decidimos cambiar de estrategia y anotamos la raíz con @Inheritance(strategy=InheritanceType.TABLE_PER_CLASS) antes de generar un nuevo esquema. ¿Es posible que se mantenga el número de tablas usadas en la base de datos para representar la jerarquía de clases? Justifique la respuesta para puntuar.

@Inheritance(strategy=InheritanceType.SINGLE_TABLE) crea una tabla única, mientras que @Inheritance(strategy=InheritanceType.TABLE_PER_CLASS) crea una tabla por cada clase concreta (no abstracta).

La única forma de que se mantenga el NÚMERO de tablas es en el caso de que las clases sean abstractas porque así no se crean una tabla para ellas.

Pregunta 10

Finalizado

Puntuación 0.10 sobre 0.10

🚩 Marcar pregunta

Ponga un ejemplo de línea de estado perteneciente a una respuesta del protocolo HTTP en la que se indica que ha habido un error interno del servidor.

```
HTTP/1.1 500 Internal Server Error
```

Pregunta 11

Finalizado

Puntuación 0.10 sobre 0.10

🚩 Marcar pregunta

¿Cuáles son las capas en que se dividen las aplicaciones empresariales de acuerdo con el estándar Jakarta (antes Java) EE?

cliente, presentación, negocio y sistema de información

Pregunta 12

Finalizado

Puntuación 0.10 sobre 0.10

🚩 Marcar pregunta

Necesitamos crear en TypeScript dos interfaces:

- Una interfaz Animal que tenga un atributo nombre de tipo cadena de caracteres.
- Una interfaz Mamifero que extienda la interfaz Animal con un atributo patas que sea numérico.

Escriba el código TypeScript para definir ambas interfaces.

LA INTERFAZ ANIMAL ES LA SIGUIENTE

```
interface Animal{  
  nombre: string;  
}
```

LA INTERFAZ MAMIFERO ES LA SIGUIENTE:

```
interface Mamifero extends Animal{  
  patas: number;  
}
```

Pregunta 13

Incorrecta

Puntúa -0,10
sobre 0,10

Desmarcar

En JPA, si todas las clases entidad de una aplicación tienen atributos anotados con `@Version` no pueden ocurrir lecturas fantasma.

- ☐ a. Falso
- ☒ b. Verdadero



Pregunta 14

Sin contestar

Valor: 0,10

Marcar
pregunta

Hemos desarrollado una aplicación Web con un *frontend* en Angular y un *backend* en Spring. Durante el desarrollo, el *backend* está disponible en `localhost:8080` y al *frontend* se accede escribiendo `localhost:4200` en el navegador. Ambos se conectan sin problema y puedo hacer uso de la aplicación.

Ejecutamos las dos aplicaciones, tal como lo hacemos en nuestra máquina de desarrollo, en un servidor llamado `si1.uma.es`, donde el *backend* sigue escuchando en el puerto 8080 y al *frontend* sigo accediendo a través del puerto 4200, pero ahora la aplicación no funciona: el *frontend* no puede conectarse al *backend*. No he cambiado nada en las aplicaciones y no hay problemas de *proxies* o *firewalls*. ¿Por qué no se conecta el *frontend* al *backend*?

Pregunta 15

Finalizado

Puntúa 0,10
sobre 0,10

Marcar
pregunta

¿Qué diferencia hay entre `var` y `let` al declarar variables en Javascript?

`var` es utilizado en el ámbito de funciones y global, mientras que `let` es utilizado en el ámbito de bloque