

## **Práctica obligatoria:**

### **404: Una aplicación web de preguntas y respuestas**

Instrucciones de entrega y criterios de corrección

#### **Instrucciones de entrega**

La fecha límite de entrega es el **jueves 16 de diciembre de 2022** (el enlace se cerrará a las **08:00** y no se admitirán entregas fuera del plazo).

Se debe subir al CV un único fichero **gxx.zip** (xx: número de grupo de laboratorio) que contenga el **proyecto Node** teniendo en cuenta los siguientes puntos:

- El fichero principal que arranca el servidor se llama **app.js**. El proyecto ha de ser arrancable mediante **node app.js**.
- Se incluye un fichero **.sql** en el directorio raíz del proyecto con las sentencias que crean la BD. Este fichero puede ser generado automáticamente por phpMyAdmin (comando export). El contenido de la base de datos se describe al final de este documento.
- Se incluye un fichero **config.js** en el directorio raíz del proyecto con los datos de configuración de la base de datos y el puerto del servidor (con la misma estructura que el fichero de mismo nombre de las partes 4 y 5 de la práctica guiada).
- **No se incluye el directorio node\_modules** pero sí los ficheros package.json y package-lock.json.

**Para el día de la defensa de la práctica**, se preparará una presentación que describa los siguientes aspectos fundamentales del trabajo:

- Diseño de la base de datos (modelo entidad-relación y modelo relacional)
- Estructura de la aplicación (controladores, routers, modelos y otros módulos)
- Listado de la rutas gestionadas por el servidor describiendo brevemente cada una de ellas.
- Descripción del método utilizado para restringir el acceso a las rutas para los usuarios no logueados.
- Gestión de los errores 404 y 500
- Listado de los paquetes utilizados (por ejemplo, express, ejs, body-parser, multer, etc) describiendo brevemente la utilidad de cada paquete.

El contenido de la presentación debe ser esquemático pero contener toda la información requerida.

## Criterios de corrección

### *Criterios eliminatorios*

Las siguientes situaciones conllevarán la calificación de NO APTO en la convocatoria actual:

- Ausencia de entrega, o entrega de una aplicación con ausencia total de funcionalidad.
- Práctica total o parcialmente copiada de otros compañeros/as, o copiada de otras fuentes. Esta situación, además, será notificada ante el Comité de actuación ante copias de la Facultad.
- Diseño de base de datos incorrecto.
- Fallos en la aplicación web que impidan el arranque del servidor.
- Imposibilidad de crear cuenta de usuario, o hacer login. Se considera una funcionalidad tan básica, que su falta impediría comprobar el funcionamiento del resto de la aplicación.
- Introducción de código SQL de acceso a base de datos dentro de los manejadores de ruta. La capa de acceso a datos debe estar implementada en módulos separados del resto de la aplicación.
- Introducción de lógica de aplicación dentro de los ficheros EJS. El código dentro de las plantillas EJS debe estar destinado únicamente a la visualización del resultado como documento HTML.

### *Criterios no eliminatorios*

Las siguientes situaciones no conllevan necesariamente a la calificación de NO APTO, salvo que concurran de manera generalizada por toda la práctica. No obstante, pueden penalizar la calificación final de la práctica.

- Problemas en el funcionamiento de la aplicación, bien porque provoquen la parada del servidor, o porque no produzcan el comportamiento esperado.
- Ausencia de validación de formularios.
- Elección incorrecta del tipo de peticiones realizadas (GET o POST). Por ejemplo, no debe utilizarse una petición GET para los formularios de login o creación de nuevo usuario.
- Uso de sentencias SQL no paramétricas, de modo que se permita la posibilidad de ataques de inyección SQL.

### *Aspectos que se valorarán positivamente*

- Separación del código fuente de la práctica en distintos módulos.
- Uso de middlewares para evitar la repetición de operaciones entre los distintos manejadores de ruta.
- Uso de subplantillas EJS para evitar repetición de código.
- Uso de routers en Express.js para dividir la funcionalidad de la aplicación en distintas subaplicaciones.

### *Contenido de la base de datos*

El script de creación de la base de datos debe incorporar además de la creación del esquema, la incorporación de los datos que se muestran en las siguientes tablas.

Los ficheros con las imágenes de perfil están disponibles en el campus.

Usuarios			
Correo	Contraseña (*)	Nombre	Imagen de perfil
nico@404.es	1234	Nico	Proporcionada por el usuario
roberto@404.es	1234	Roberto	Proporcionada por el usuario
sfg@404.es	1234	SFG	Proporcionada por el usuario
marta@404.es	1234	Marta	Proporcionada por el usuario
lucas@404.es	1234	Lucas	Proporcionada por el sistema
emy@404.es	1234	Emy	Proporcionada por el usuario

(\*) Si la aplicación impone un patrón para la contraseña se puede cambiar la propuesta en la tabla.

Pregunta	
Título	¿Cual es la diferencia entre position: relative, position: absolute y position: fixed?
Cuerpo	Sé que estas propiedades de CSS sirven para posicionar un elemento dentro de la página. Sé que estas propiedades de CSS sirven para posicionar un elemento dentro de la página.
Etiquetas	css y css3
Autor	nico@404.es
Respuesta	
Texto	<p>La propiedad position sirve para posicionar un elemento dentro de la página. Sin embargo, dependiendo de cual sea la propiedad que usemos, el elemento tomará una referencia u otra para posicionarse respecto a ella.</p> <p>Los posibles valores que puede adoptar la propiedad position son: static   relative   absolute   fixed   inherit   initial.</p>
Autor	lucas@404.es

Pregunta	
Título	¿Cómo funciona exactamente nth-child?

Cuerpo	No acabo de comprender muy bien que hace exactamente y qué usos prácticos puede tener.
Etiquetas	css y html
Autor	roberto@404.es
<b>Respuesta</b>	
Texto	La pseudoclase :nth-child() selecciona los hermanos que cumplan cierta condición definida en la fórmula $an + b$ . a y b deben ser números enteros, n es un contador. El grupo an representa un ciclo, cada cuantos elementos se repite; b indica desde donde empezamos a contar.
Autor	emy@404.es

<b>Pregunta</b>	
Título	Diferencias entre == y === (comparaciones en JavaScript)
Cuerpo	<p>Siempre he visto que en JavaScript hay:</p> <pre>asignaciones = comparaciones == y ===</pre> <p>Creo entender que == hace algo parecido a comparar el valor de la variable y el === también compara el tipo (como un equals de java).</p>
Etiquetas	JavaScript
Autor	sfg@404.es

<b>Pregunta</b>	
Título	Problema con asincronismo en Node
Cuerpo	<p>Soy nueva en Node... Tengo una modulo que conecta a una BD de postgres por medio de pg-node. En eso no tengo problemas. Mi problema es que al llamar a ese modulo, desde otro modulo, y despues querer usar los datos que salieron de la BD me dice undefined... Estoy casi seguro que es porque la conexion a la BD devuelve una promesa, y los datos no estan disponibles al momento de usarlos.</p>
Etiquetas	nodejs
Autor	marta@404.es

Pregunta	
Título	¿Qué es la inyección SQL y cómo puedo evitarla?
Cuerpo	<p>He encontrado bastantes preguntas en StackOverflow sobre programas o formularios web que guardan información en una base de datos (especialmente en PHP y MySQL) y que contienen graves problemas de seguridad relacionados principalmente con la inyección SQL.</p> <p>Normalmente dejo un comentario y/o un enlace a una referencia externa, pero un comentario no da mucho espacio para mucho y sería positivo que hubiera una referencia interna en SOes sobre el tema así que decidí escribir esta pregunta.</p>
Etiquetas	mysql y sql
Autor	lucas@404.es

Además de los usuarios, las preguntas y respuestas, **la base de datos no debe tener registro ni de visitas, ni de votaciones** y como consecuencia de ello los usuarios en su perfil no tendrán ni reputación ni medallas aunque sí número de preguntas y número de respuestas.