

Examen CORE 2019-2020 - 2º Parcial - Práctico

Preparación Inicial

El alumno debe aplicar las migraciones y seeders del proyecto ejecutando los comandos:

```
$ npm run migrate  
$ npm run seed
```

(o migrate_win y seed_win si utiliza Windows).

Enunciado

Este examen consiste en añadir un botón en la barra de navegación que muestre el número total de quizzes que hay en la tabla Quizzes de la base de datos.

Para ello hay que crear un botón en la barra de navegación, definir la nueva ruta que invoca el botón, crear una función middleware que atienda la petición, y crear una vista EJS que muestre los resultados.

Además, esta funcionalidad debe estar disponible solo para usuarios logueados.

Paso 1:

Crear un botón con el texto "**Contar**" en views/layout.ejs, que al pulsarlo envíe una petición HTTP con el método GET y el path `"/quizzes/contar"`.

Paso 2:

Definir en routes/index.js la ruta GET + `"/quizzes/contar"`. Esta ruta debe ser atendida por una función middleware llamada **contar** que se exporta desde controllers/quiz.js.

Paso 3:

Editar controllers/quiz.js para implementar y exportar el middleware llamado **contar**.

Este middleware calculará el número total de quizzes que hay en la tabla Quizzes, y los mostrará renderizando la vista views/quizzes/examen.ejs.

Para contar los quizzes que hay en la tabla Quizzes debe invocar la promesa:

```
models.Quiz.count()
```

que se resuelve devolviendo el número de quizzes contados.

Para renderizar la vista anterior debe usar la funcion `res.render`, y debe pasar el número de quizzes contados en un parámetro o variable llamado total.

Paso 4:

Implementar la vista `views/quizzes/examen.ejs`. A esta vista se le pasa un parámetro llamado `total` que contiene el valor a mostrar.

Esta vista mostrará una cabecera `<h1>` con el texto "**Quizzes Contados**", y un párrafo `<p>` con el texto "Hay XXX quizzes", donde XXX debe sustituirse por el valor de la variable `total`.

Paso 5:

Modificar el paso 1 para que el botón "**Contar**" solo se muestre si el usuario se ha logueado.

Paso 6:

Modificar el paso 2 para que el middleware **contar** solo se ejecute si el usuario se ha logueado.

Prueba y entrega del examen

Este examen incorpora un autocorrector que prueba el desarrollo pedido en el examen. Para ejecutar el autocorrector debe ejecutar el comando:

```
$ npx autocorrector
```

Puede ejecutar el autocorrector tantas veces como se desee sin ninguna repercusión en la calificación.

Recuerde que, para entregar el examen, el alumno debe subirlo a Moodle usando el autocorrector. Para ello debe ejecutar el comando:

```
$ npx autocorrector --upload
```

MUY IMPORTANTE: Si no se suben los resultados usando el autocorrector al menos una vez, la entrega se suspende automáticamente.