**Bootcamp: Tecnologías para el desarrollo web**

**Master en Full Stack Web Development**

**Presentación de la actividad y objetivos**

Con este examen, y tomando referencia todas las competencias detalladas en la guía del módulo, se evaluará el nivel de conocimientos del alumno tras haber cursado el Curso nivelador del Master en Full Stack Web Development Threepoints.

**Metodología**

La prueba consta de varios apartados donde se pide al alumno que desarrolle desde cero un proyecto sencillo de Nodejs para, continuación, dockerizar la solución. Todo esto siguiendo un sencillo control de versiones del proyecto haciendo uso de Git y GitHub.

**Criterios de evaluación**

Esta prueba **se puntuará sobre 10 puntos**. Para poder aprobar el examen se deberá sacar **una nota igual o superior a un 5.0**.

**Ejercicios a realizar**

1. **Instalar un IDE en vuestra máquina**. De acuerdo con las preferencias del profesorado del Máster, recomienda Visual Studio Code, pero se permite flexibilidad en la elección. (*0.5 puntos*)

Se realiza instalación de Visual Code para realizar las actividades dentro de la tarea.

1. **Crearse un perfil de *GitHub***. (*1.0 puntos*)

Se crea cuenta de GitHub donde realiza fork del repositorio entregado por el profesor y se crea nueva rama con el nombre tarea para realizar los cambios necesarios.

<https://github.com/ALBCAB03/Bootcamp-Full-Stack/tree/Tarea>

1. **Desplegar una base de datos de MongoDB en un container de Docker con persistencia.**
2. Crear una database y una colección o colecciones para almacenar documentos. (2 puntos) Crear algunos documentos en la(s) colección(es) que se ha(n) creado. Pueden tener los mismos campos que los que se vieron en clase o ser de un origen totalmente distinto.

Se crea base de datos siguiendo los pasos en la guíaburros y utilizando los comandos, crear una red

docker network create mynetwork

para luego crear un contenedor con persistencia

PS D:\Alberto\Estudio\Inesdi-Full Stack Developer\Repositorios\Bootcamp-Full-Stack> docker run --name mongodb-container --hostname mongodb-container -d -p 27017:27017 --network mynetwork -v \db\_sample\users.json mongo

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Se puede rellenar la base de datos con mecanismo de llamada a alguna API o se pueden meter a mano o facilitar mockeados. Por ejemplo: https://github.com/eblancoh/Bootcamp-Full-Stack/blob/main/db\_sample/users.json

Utilizando MongoDB compas y los datos planteados en clase se procede a rellenar la base de datos.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. **Crear una API sencilla con Nodejs (3 puntos) que sea capaz de:**
2. Conectarse al mongo dockerizado. Crear una ruta que, mediante un GET, devuelva todos los documentos en una colección

De igual forma para conectarse al mongo dockerizado siguiendo los pasos en la guía

Texto

Descripción generada automáticamente

1. Definir una ruta que, mediante un GET, devuelva sólo los documentos que cumplen la condición en base a un query sobre uno o varios campos de un documento.
2. Definir una ruta con método PUT que modifique alguno de los campos de un documento (el cual debe cumplir alguna condición o query). Los códigos 4xx se dejan a su elección. Los códigos 2xx deben ser, como mínimo, los siguientes:

* If not found, create a new document in the database. (return 201 Created)
* If found, target keyword(s) to be successfully modified (200 OK)

1. Definir una ruta con método DELETE que elimine el(los) documento(s) que cumplan alguna condición o query. Los códigos 4xx se dejan a su elección. Los códigos 2xx deben ser, como mínimo, los siguientes:

* If not found, do nothing. (204 No Content)
* If found, document deleted (200 OK)

1. **Dockerizar la API del punto 5**. (1 punto)

Para dockerizar la aplicación se

Texto

Descripción generada automáticamente

1. **Subir a un repo público de GitHub el repositorio creado para realizar el punto 5 y 6**. El repositorio debe ir acompañado de un *README.md* que contenga una descripción e instrucciones para poder usar el código en otra máquina. (*2.5 puntos*)
   * 1. - Se pide que el alumno intente trabajar haciendo uso de, al menos, una rama además de la main/master.
     2. - De esta manera, se pide que se realice al menos una Pull Request mínimamente documentada y un mergeo a main antes de entregar la actividad. Por ello, también se pide no borrar la rama tras el mergeo.