Prueba técnica ALMUNDO FullStack

Ejercicio 1: API Rest NodeJS

Este punto consiste en armar un API REST en Node.js. El mismo será utilizado en el ejercicio 2.

API, Node, Express, MongoDB, Mongoose.

* La funcionalidad de listado y filtrado de hoteles debe estar soportada por la API y consumida en la aplicación cliente

Se creó el servicio rest con filtro por estrellas y nombre

* A la hora de diseñar la estructura de la aplicación, tener en cuenta factores como escalabilidad, reutilización y separación de responsabilidades.

Se definen paquetes controllers, models y se designan responsabilidades a cada paquete tal que se promueva la mantenibilidad

* Uso de configuraciones para ajustar como se ejecuta la aplicación en entornos productivos y de desarrollo.

Se utilizan archivos de configuraciones para definir la base de datos a utilizar, el puerto y demás, esto con ayuda del plugin config

Ejercicio 2: Frontend

Angular 2, Bootstrap

* Maquetar una página de resultado de hoteles, ver imágenes en el repo (solo mobile y desktop).

OK, Se maqueto desde 0 creando componentes según la necesidad

* Consumir la API desarrollada en el ejercicio anterior, implementando las funcionalidades necesarias para listar y filtar los hoteles.

OK, Se consume el api creado con anterioridad

* Utilizar alguna librería o framework guiado por componentes ( AngularJS o Angular)

OK, Angular (Angular2) Dado que tiene mayor soporte y la proyección de google es que sea retroactivo, Angular 1 está siendo abandonado.

* Utilizar algún package manager (npm, bower) para manejar dependencias externas.

OK, Se usa npm

* Utilizar una grilla responsive o similar para el maquetado.

OK, Se usa Bootrap

* Optimizar todos los recursos para entornos productivos, (minificar, ofuscar, etc).

OK, Angular.cli Dispone de configuraciones establecidas respecto a dependencias, minifación y ofuscación para archivos js y css

Extras

Los puntos extras solo se tomarán en cuenta si las funcionalidades de los puntos anteriores

Fueron completadas correctamente.

* Añadir alguna capa de persistencia de datos.

Se montó la base de datos local, mongodb

* Implementar el CRUD de hoteles (solo a nivel API).

Se implementaron los servicios rest para CRUD según convenciones, plural, get, post, put, delete

<http://www.restapitutorial.com/lessons/restfulresourcenaming.html>