



UNAH

UNIVERSIDAD NACIONAL
AUTÓNOMA DE HONDURAS

DEPARTAMENTO DE
INGENIERÍA EN SISTEMAS

Programación Orientada a Objetos

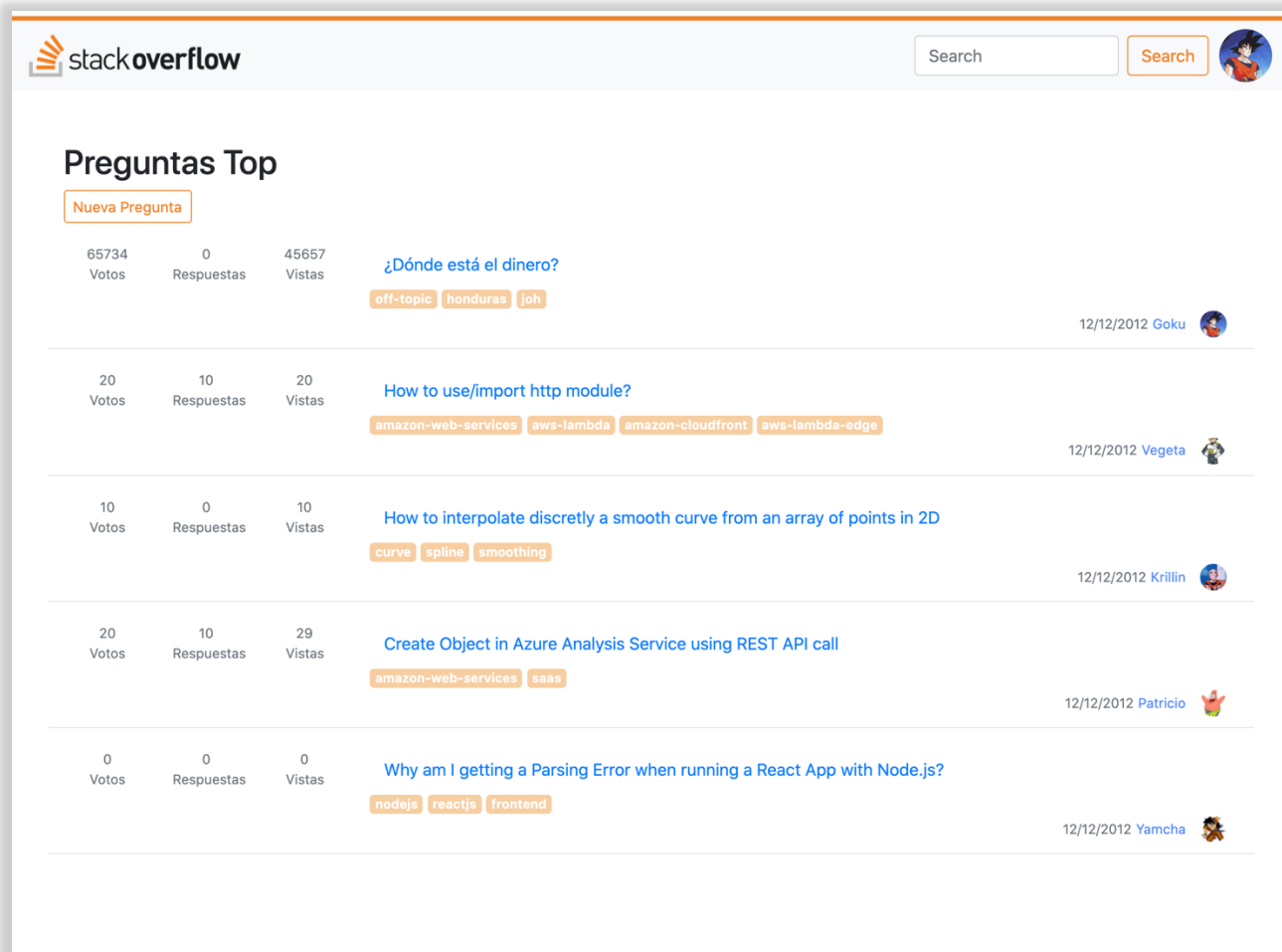
Catedrático: Erick Vladimir Marín

La presente evaluación es de carácter individual, evite cualquier tipo de plagio o será penalizado.

El objetivo de esta evaluación es aplicar conceptos de frontend con Angular y backend con NodeJS/MongoDB tomando como referencia la plataforma web que salva muchas vidas de programadores... **StackOverflow**.

Se le provee

Frontend HTML, CSS, Bootstrap: Se provee una funcionalidad básica para mostrar/ocultar regiones y para mostrar ventanas modales.



En base a los recursos proveídos desarrollar lo siguiente:

- **Crear base de datos en MongoDB:** Crear una base de datos con el nombre **stackoverflow** y dos colecciones; **usuarios** y **preguntas**. La estructura de los documentos a guardar se describe en el apartado posterior.
- **Cambiar usuario:** En la esquina superior derecha, sobre la imagen del usuario, se debe dar click y se visualizará una ventana modal con la lista de los usuarios registrados. Se debe agregar la lógica necesaria para listar los usuarios y poder seleccionar uno en particular. Esta sección simulará un login para identificar cual es el usuario actual. Este usuario filtrado se utilizará para agregar preguntas o respuestas.
- **Renderizar la lista de preguntas:** Como se visualiza en la imagen anterior, para cada pregunta se debe visualizar la cantidad de votos, cantidad de respuestas, cantidad de visualizaciones, el título de la pregunta, los hashtags y los datos del usuario



UNAH

UNIVERSIDAD NACIONAL
AUTÓNOMA DE HONDURAS

DEPARTAMENTO DE
INGENIERÍA EN SISTEMAS

Programación Orientada a Objetos

Catedrático: Erick Vladimir Marín

que hizo la pregunta. Independientemente del usuario seleccionado, se deben visualizar todas las preguntas.

- **Funcionalidad para agregar una nueva pregunta:** al dar click en el botón Nueva Pregunta se mostrará una ventana modal para registrar la pregunta y los hashtags o palabras clave separadas por coma. La pregunta deberá guardarse en el listado de preguntas, el usuario que figura como dueño de la pregunta es el seleccionado de la lista de usuarios.
- **Renderizar el detalle de una pregunta en particular:** Se debe visualizar el detalle de una pregunta al momento de dar click sobre el título de ella en la lista de preguntas. El detalle de la pregunta se muestra a continuación:

stackoverflow

Search

Search

How to use/import http module?

45

I've been playing with Angular 2 Quickstart.
How can I use/import http module in Angular 2?
I've looked at Angular 2 Todo's.js, but it doesn't use the http module.
I've added "ngHttp": "angular/http", to dependencies in package.json because I've heard Angular 2 is somewhat modular.

angular angular-http modules

12/12/2012
Majimboo

2 Respuestas

105

In version 37 you need to do this:

```
/// >  
import {Http} from "angular2/http";  
And run this tsd command:  
tsd install angular2/http
```

12/12/2012
Picoro

10

In version 37 you need to do this:

```
import {Http} from "angular2/http";
```

Votación: En la parte superior se debe visualizar el título, descripción de la pregunta, los datos del usuario, un indicador de la cantidad de votos y la lista de los hashtags. Se debe dar funcionalidad a los botones flecha arriba ▲ para incrementar un voto y flecha abajo ▼ para decrementar un voto.



Renderizar Respuestas: Inmediatamente después del detalle de la pregunta se debe visualizar las respuestas en un formato muy similar a las preguntas. Al igual que las preguntas, las respuestas también tienen votos y se pueden incrementar o decrementar. Entre más votos tenga una respuesta deberá aparecer primero.

Estructura de la información:

Para gestionar la información desde el backend deberá crear dos colecciones, uno para registrar usuarios y otro para registrar las preguntas con sus respectivas respuestas.

En la colección **usuarios** se deben registrar objetos como el mostrado a continuación. En el caso del atributo preguntas, únicamente es para mantener una referencia de las preguntas hechas por el usuario, sin embargo, la información completa de cada pregunta estará en la colección de preguntas.

```
{
  id: '1',
  nombre: 'Goku',
  urlImagen: 'img/profile-pics/goku.jpg',
  preguntas: [
    {
      id: '1',
      título: '¿Dónde está el dinero?'
    }
  ]
}
```

En la colección de preguntas, deberá registrar todas las preguntas con su detalle en un arreglo de JSONs, cada JSON deberá tener una estructura como se muestra a continuación, en caso del atributo hashtags deberá ser un arreglo de cadenas, un ítem por cada hashtag:

```
{
  id: '1',
  título: '¿Dónde está el dinero?',
  descripción: 'Lorem ipsum dolor sit.',
  fecha: '12/12/2012',
  votos: 5,
  vistas: 10,
  hashtags: ['off-topic', 'pn'],
  idUsuario: '1',
  respuestas: [
    {
      id: '1',
      descripción: 'Se lo robaron',
      fecha: '12/12/2012',
      votos: '56326',
      idUsuario: '1'
    }
  ]
}
```



Observaciones:

- Cree solo los servicios web del API Rest a utilizar, desarrolle los que sean necesarios para la solución.
- Estructure el proyecto en carpetas independientes para frontend y backend.
- Agregue unos cuantos documentos de prueba a la base de datos. (pocos).
- Genere los ids de las colecciones utilizando la función de mongoDB.
- No modifique el nombre de la base de datos, las colecciones y los nombres de los atributos de los objetos JSON ya que puede ocasionar malentendidos al momento de revisar su evaluación.
- En el caso de los textos que se ingresen para las preguntas y respuestas se puede observar ciertas regiones en gris al momento de visualizar el detalle de la pregunta/respuestas, esas se visualizan de esa forma automáticamente si en el texto aparece algo con la etiqueta <pre>.
- No se permite el uso de otros frameworks no vistos en clase.
- Puede utilizar sus apuntes, google y cualquier recurso, escriba su código y evite copiar y pegar (eso no es programar).
- Al momento de subir el proyecto solución, no incluya la carpeta node_modules. Incluya un pequeño script básico para la base de datos.

Recomendaciones:

- Para cada acción verifique la estructura del json a utilizar y sus diferentes atributos.
- Enfóquese en su evaluación, aproveche el tiempo y no confunda a sus compañeros.
- Evite copiar código compartido en chats grupales, se penalizará exámenes donde se compruebe copia o plagio (ejem. Códigos similares con solo cambios de nombres de variables, funciones, orden del código, entre otros).

*“El éxito no llega solo..., ¡hay que romperse la madre para lograrlo!” . _ Yo
¡Éxito!*