

Entrega Hito 2 - Proyecto Final

Nombre del Sistema:
Assist Me

Entregado a:
Juan Manuel Reyes García
Christian Esteban Sanchez Pineda

Proyecto Integrador I
Proyecto del Curso 2019-1
Departamento TIC - Facultad de Ingeniería



Método de la Ingeniería

Identificación del Problema

Descripción del contexto

Globant S.A. es una compañía de TI dedicada al desarrollo y mantención de Software. Actualmente, cuenta con un amplio portafolio de servicios, entre los que se encuentran: consultorías, automatización de procesos, Big Data, desarrollo de aplicaciones, entre muchos más.

Debido a la gran cantidad de servicios que prestan, la compañía se divide en diferentes unidades organizacionales llamadas “*Studios*”, los cuales cuentan con un correo por studio (ej: bigdata@globant.com , gaming@globant.com , cybersecurity@globant.com). Estos correos permiten que cualquier duda se comunique directamente con el *Studio*, que se supone tiene mayor conocimiento sobre cierto tema. Sin embargo, se ha observado, que por este medio, se hacen ciertas preguntas que podrían ser de interés general para muchos Globers y el conjunto de todas esas preguntas podrían ser un recurso más para el personal. Reenviar estos correos, archivar los que se consideren importantes, y buscar las personas que también puedan saber sobre el tema resulta bastante dispendioso para cada *Studio*.

Por todo lo anterior, la compañía Globant tiene la iniciativa de desarrollar una aplicación web de preguntas y respuestas para el desarrollo interno de la compañía, que cuente con dos roles, uno de usuario y otro de administrador. Esta aplicación le permitirá a los Globers antiguos y a nuevos integrantes de la organización, tener acceso a una herramienta donde pueda retroalimentar y ser retroalimentados con las preguntas de los diferentes miembros de la compañía.

Problema:

Las preguntas realizadas a los *studios* de Globant no son de fácil acceso para las personas interesadas y responder a todas ellas resulta ser una tarea dispendiosa para cada *studio*.

Identificación de necesidades

La identificación de las necesidades y la especificación de los requerimientos se encuentran anexadas al final del documento.

Recopilación de la información

Con el fin de abordar la situación problema de la mejor manera, se presenta un marco teórico que se considera relevante para el contexto de este. Así mismo, el estado del arte presentado el cual permite ver soluciones realizadas en contextos similares al problema.

Librerías de Google para iniciar sesión:

En primer lugar, las librerías o Api de Google es un conjunto de API desarrollada por Google, las cuales permiten la comunicación e integración de los Servicios de Google con otros servicios. Ejemplos incluyen las API de Búsqueda, Gmail, Traductor o Maps. Las aplicaciones de terceros pueden usar esas API para extender la funcionalidad de sus servicios. Ahora bien, a veces, queremos

que los usuarios inicien sesión utilizando sus credenciales existentes de aplicaciones de terceros, como Facebook, Twitter, Google, etc. Por lo tanto, Google ofrece una API de administración (autorización) que se descompone de la siguiente manera:

1. Autorizar solicitudes

La aplicación que deseamos desarrollar debería utilizar OAuth 2.0 para autorizar las solicitudes. No se admitiría ningún otro protocolo de autorización. Sin embargo, se usa el inicio de sesión de Google, algunos aspectos de la autorización se gestionan automáticamente.

2. Autorizar solicitudes con OAuth 2.0

Este es el punto clave, ya que, todas las solicitudes a la API de Analytics deben estar autorizadas por un usuario autenticado. Sin embargo, se debe tener en cuenta que los detalles del proceso de autorización para OAuth 2.0, o "flujo", varían ligeramente dependiendo del tipo de aplicación que se esté creando. El siguiente proceso general se aplica a todos los tipos de aplicación:

1. Cuando se crea la aplicación, se debe registrar con la consola de APIs de Google. A continuación, Google proporciona la información que necesitará posteriormente, como un ID de cliente y un secreto de cliente.
2. Activar la API de Analytics en la consola de APIs de Google. Si la API no aparece en la lista de la consola de APIs, se salta este paso.
3. Cuando la aplicación necesite acceder a los datos del usuario, esta solicitará a Google un determinado alcance de acceso.
4. Google muestra una pantalla de consentimiento al usuario pidiéndole su autorización para que la aplicación solicite algunos de sus datos.
5. Si el usuario aprueba esta petición, Google proporcionará a la aplicación un token de acceso de corta duración.
6. La aplicación solicita los datos del usuario y adjunta el token de acceso a la solicitud.
7. Si Google determina que tu solicitud y el token son válidos, muestra los datos solicitados.

En algunos flujos se incluyen pasos adicionales, como el uso de tokens de actualización para adquirir nuevos tokens de acceso. También se debe tener en cuenta que para solicitar acceso utilizando OAuth 2.0, la aplicación necesita la información del alcance, así como la información que proporciona Google durante el registro de la aplicación (como el ID y el secreto de cliente).

3. Flujos de OAuth 2.0 habituales

A continuación, se enumeran los casos prácticos habituales de determinados flujos de OAuth 2.0:

- Servidor web

Este flujo resulta adecuado para el acceso automático, sin conexión o programado de los datos de Google Analytics de un usuario.

Ejemplo: Actualizar automáticamente los paneles de usuario con los datos de Google Analytics más recientes.

- Cliente

Resulta ideal en el caso de que los usuarios interactúen directamente con la aplicación para acceder a sus datos de Google Analytics desde un navegador. Con este flujo no se necesitan funciones en el servidor, pero no se pueden elaborar informes automatizados, sin conexión o programados.

Ejemplo: Una herramienta de creación de informes basada en navegador, como el Explorador de solicitudes de Google Analytics.

- Aplicaciones instaladas:

Este flujo es para aplicaciones que se distribuyen en paquetes y que instala el usuario. Es necesario que la aplicación o el usuario tenga acceso a un navegador para completar el flujo de autenticación.

Ejemplos:

- Un widget de ordenador en un PC o Mac.
- Un complemento de un sistema de gestión de contenido. La ventaja de este flujo, en relación con el de servidor web o de cliente, es que se puede usar un solo proyecto de la consola de APIs para la aplicación. De este modo, se pueden elaborar informes consolidados y se simplifica la instalación para los usuarios.

- Cuentas de servicio

Las cuentas de servicio son útiles para el acceso automatizado, sin conexión o programado a los datos de Google Analytics de tu propia cuenta. Por ejemplo, para crear un panel activo de tus propios datos de Google Analytics y compartirlo con otros usuarios.

4. Solucionar problemas

No se otorgará la autorización en los casos siguientes:

- Se muestra el código de estado 401 si tu access_token ha caducado o si utilizas un alcance erróneo para la API.
- Se muestra el código de estado 403 si el usuario autorizado no tiene acceso a la vista (perfil). Asegúrate de estar autorizado con el usuario correcto y que este tenga la vista (perfil) que has seleccionado.

5. Espacio de OAuth 2.0

Esta herramienta permite llevar a cabo todo el proceso de autorización mediante una interfaz web. En la herramienta también se muestran todos los encabezados de solicitud HTTP necesarios para realizar solicitud una autorizada. Si no se consigue que la autorización funcione en la aplicación, se debe probar que lo haga mediante el espacio de OAuth 2.0. Después, puedes comparar los encabezados HTTP y enviar una solicitud al espacio sobre lo que tu aplicación envía a Google Analytics. Esta comprobación es una forma sencilla de garantizar que se aplica el formato correcto a tus solicitudes.

Concesión no válida:

Cuando se intenta utilizar un token de actualización, se obtiene un error `invalid_grant` en los casos siguientes:

- El reloj de tu servidor no está sincronizado con el protocolo NTP.
- Se ha superado el límite de tokens de actualización.

Las aplicaciones pueden solicitar varios tokens de actualización para acceder a una sola cuenta de Google Analytics.

Por ejemplo, si un usuario quiere instalar una aplicación en varios ordenadores y acceder a la misma cuenta de Google Analytics, necesitará un token distinto para cada ordenador. Cuando el número de tokens de actualización supera el límite, los tokens anteriores dejan de ser válidos. Si la aplicación intenta utilizar un token de actualización invalidado, se devuelve la respuesta de error `invalid_grant`.

El límite para cada par único de cliente OAuth 2.0 y cuenta de Google Analytics es de 25 tokens de actualización. Si la aplicación sigue solicitando tokens de actualización para el mismo par de cliente y cuenta, cuando se solicita el token número 26, el primer token de actualización que se emitió anteriormente deja de ser válido. El token de actualización número 27 que se solicite invalidará el segundo token emitido anteriormente, y así sucesivamente.

https://es.wikipedia.org/wiki/API_de_Google

<https://www.c-sharpcorner.com/article/authentication-using-google-in-asp-net-core-2-0/>

<https://developers.google.com/analytics/devguides/config/mgmt/v3/authorization?hl=es>

Bases de datos:

Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

<http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>

Microsoft SQL Server:

Es un sistema de gestión de base de datos relacional desarrollado por Microsoft. Como un servidor de base de datos, es un producto de software con la función principal de almacenar y recuperar datos

según lo solicitado por otras aplicaciones de software que pueden funcionar ya sea en el mismo ordenador o en otro ordenador a través de una red.

Diseño responsive:

Diseño cuyo objetivo es adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visitarlas. Cada dispositivo tiene sus características concretas y esta tecnología pretende que con un único diseño web, todo se vea correctamente en cualquiera de estos. (Tomado de https://es.wikipedia.org/wiki/Dise%C3%B1o_web_adaptable)

Analítica de datos:

Es una ciencia que tiene como propósito tomar un conjunto de datos y sacar conclusiones sobre ellos. Normalmente se divide en tres ramas: el análisis exploratorio de datos, para descubrir nuevas herramientas de los datos; el análisis confirmatorio, para confirmar teorías sobre un grupo de datos; y el análisis cualitativo donde se examinan datos de tipo no numérico para inferir conclusiones.

<https://searchdatacenter.techtarget.com/es/definicion/Analisis-de-Datos>

Función autocompletar de Google:

A la hora de realizar una búsqueda en Google automáticamente -y en tiempo real- la plataforma sugiere opciones de palabras o frases de las cuales se puede tratar la búsqueda en cuestión. Esta funcionalidad de autocompletado se realiza con el fin de reducir el tiempo de teclado y se basa en las siguientes fuentes de predicción:

- Términos que se ingresan. Esto ya ocurre cuando se ha ingresado una palabra.
- Búsquedas relevantes realizadas anteriormente.
- Búsquedas de otras personas que se volvieron populares en el radar (Tendencias).

Las predicciones se realizan automáticamente por medio de un algoritmo que depende de los términos descritos anteriormente.

<https://support.google.com/websearch/answer/106230?co=GENIE.Platform%3DAndroid&hl=es-419>

Adicionalmente, la funcionalidad de autocompletar tiene políticas sobre sus sugerencias tales como: predicciones sexuales, incitación al odio hacia alguna persona, acciones violentas o de incitación criminal.

<https://support.google.com/websearch/answer/7368877>

Cuales son los algoritmos más usados para hallar textos similares

Para textos cortos los más recomendados son Td-idf o BM-25 para pesar, y luego usar similaridad por coseno.

Para textos largos se recomienda usar “Locality-Sensitive Hashing”.

Tf-idf (Term frequency - inverse document frequency)

Es una medida numérica que expresa la relevancia de una palabra en un documento. Este valor crece proporcionalmente a la cantidad de veces que se repite la palabra en el documento pero es

compensada por la cantidad de veces que aparece la palabra en toda la colección de documentos, esto con el fin de manejar el hecho de que algunas palabras son más comunes que otras.

Okapi BM25

Es una función que sirve para clasificar documentos semejantes de acuerdo a su relevancia en una consulta dada. Está basado en un modelo de trabajo probabilístico.

BM25 está basado en el concepto de bolsa de palabras, con esto se representan los documentos que se desean ordenar en función de su relevancia con una consulta dada.

Por lo que tenemos que, dada una consulta Q , que contiene las palabras claves q_1, q_2, \dots, q_n el valor de relevancia para un documento D será,

$$\text{score}(D, Q) = \sum_{i=1}^n \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)},$$

donde $f(q_i, D)$ es la frecuencia con la que aparecen las palabras clave en el documento D . $|D|$ es la longitud del documento en palabras y avgdl es la longitud promedio de la colección de documentos. b y k_1 son parámetros que suelen depender de las características de la colección, aunque normalmente se le asignan valores $k_1 = 1.25$ y $b = 0.75$.

$\text{IDF}(q_i)$ es el peso de las palabras clave que aparecen en la consulta y se calcula de la siguiente manera,

$$\text{IDF}(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}$$

Donde N es el número de documentos en la colección y $n(q_i)$ es el número de documentos que contienen la palabra clave q_i .

Extraer palabras clave de un texto corto

El producto de extraer palabras claves de un texto corto se puede lograr mediante el uso de machine learning. De acuerdo al autor del texto “Extracting Keywords From Short Text”, expuso tres opciones distintas como soluciones:

1. Estilo Scrabble: se toma la oración y se calcula cada valor de cada palabra como en el juego de Scrabble y se retornan las mayor puntuadas como las más importantes. Luego se toman las palabras de la siguiente oración y se compara con las anteriores para buscar aquellas palabras relacionadas o repetidas. En este caso, las palabras con mayores puntajes tienden a ser las más importantes y las que más se repiten excepto por ciertos casos; para evitar esto se implementa una lista que contenga este tipo de palabras consideradas como excepciones y no contarlas en el ranking. Al finalizar, el ranking debe de tener una lista de puntajes en el cual las palabras más importantes estarían de primeras.

2. Método RAKE (Rapid Automated Keyword Extraction): Este método consiste en eliminar las palabras más comunes en el idioma como conectores entre otros para al final dejar las palabras que se podrían considerar no comunes o las importantes del texto. En este caso no se garantiza perfectamente que sean las palabras más importantes del texto corto. Por último, RAKE se encuentra implementado en varios lenguajes de programación lo cual lo hace bastante simple y sencillo de usar.
3. Inventar un nuevo Lenguaje: En este caso sería crear un lenguaje mediante el uso de palabras importantes como secuencia y hacer que la máquina aprenda a reconocerlas mediante una base de datos, tomándolas y creando un nuevo lenguaje basándose en ellas. Primero es necesario generar un set de datos, en el cual se pueda relacionar las palabras claves a algún tema y así empezar a categorizarlas. Luego, se cargan los datos del programa en la red neuronal con una estructura base. Para continuar, se ejecuta la red neuronal y se deja entrenando la secuencia de secuencias hasta que termine, al terminar la red neuronal ya debería de tener la capacidad de identificar algunas o varias palabras claves dentro del texto que se ingrese. Para mejores resultados es preferible que se entrene la red neuronal con una cantidad de datos mayor.
4. De acuerdo a la página *Elastic*, en el artículo titulado “More Like this Query” o Consulta Mas Como Esto, consiste en encontrar documentos parecidos en base a un conjunto de documentos mediante la búsqueda de términos similares dentro de los documentos. Para lograr este proceso, se seleccionan una cantidad representativa de términos en el documento ingresado, y luego se genera una petición de búsqueda en la base de datos para encontrar textos que contengan términos similares. Para que esta consulta funcione, primero se debe de clasificar el texto ingresado dentro de un tipo de consulta, en este caso se utiliza como método la fórmula de puntaje de Lucene (buscar similitudes dentro del texto, asignarles un puntaje y comparar donde pertenece). Luego, se toma el texto del documento ingresado y se analiza para poder extraer n cantidad de términos clave y utilizarlos en la consulta para la búsqueda de textos similares. Al finalizar el proceso, se retorna un conjunto de documentos similares al que se ingresó.

API's para extraer palabras claves

- Keywords Extraction API Documentation de Rapid Api: Esta API permite tomar palabras clave de uno o varios URL o textos. Como parámetros de entrada pide un string de JSON especificando la entrada como tal de textos (el texto plano o el url), seguido del tipo de entrada y el número de palabras clave a encontrar. Cabe resaltar que para esta API la versión gratuita admite solo hasta 100 requests, y es necesario meter tarjeta.

▼ Header Parameters

X-RapidAPI-Key

SIGN-UP-FOR-KEY

STRING

REQUIRED

▼ Required Parameters

Input Format

JSON_STRING

```

{
  "input_data":
    ["https://en.wikipedia.org/wiki/Donald_Trump",
     "https://en.wikipedia.org/wiki/Hillary_Clinton"],
  "input_type": "url",
  "N": 10
}

```

REQUIRED

1) input_type = "url" or "text", 2) input_data = ["list of text or url strings"], 3) N = number of keywords required

Sample Response

```

{
  "status": "OK",
  "message": "Keywords extracted successfully",
  "result": [ 5 items
    0 : "keyword 1"
    1 : "keyword 2"
    2 : "keyword 3"
    3 : "keyword 4"
    4 : "keyword 5"
  ]
}

```

Imagen tomada de la documentación de Rapid Api para extracción de palabras clave <https://rapidapi.com/UnFound/api/keywords-extraction2>

Estado del arte:

Se han identificado dos plataformas web que tienen funcionalidades parecidas a las solicitadas, como lo es StackOverflow y Quora, cuya estructura será explicada y se tomará como base para realizar la interfaz gráfica.

StackOverflow :

Stack Overflow es un sitio de preguntas y respuestas para programadores y entusiastas del desarrollo y uso de software. Hace parte de la red de sitios de preguntas y respuestas de Stack Exchange. (Tomado de <https://es.stackoverflow.com/tour>)

stackoverflow

en español

Buscar...

Inicio sesión

Registrarse

Inicio

Preguntas

Etiquetas

Usuarios

Explora nuestras preguntas

php

javascript

java

html

android

c#

jquery

python

css

android-studio

más etiquetas

Activo

Destacadas

Popular

Semana

Mes

1 voto

0 respuestas

2 vistas

Laravel problema con paginación

laravel

paginación

formulada hace 43 segundos Laura1995 71

1 voto

2 respuestas

11 vistas

¿Como agregar caracteres a una palabra contenida en un texto?

javascript

html

jquery

html5

modificada hace 2 minutos KacosPro 2,027

0 votos

1 respuesta

17 vistas

Subir una imagen, guardarla y mostrarla durante la sesion como imagen de perfil con PHP

php

fileupload

sesiones

respondida hace 9 minutos Mario Guiber 1,239

0 votos

0 respuestas

5 vistas

Lista doblemente enlazada agregar delante y agregar detrás C#

c#

formulada hace 20 minutos Kevin 1

0 votos

0 respuestas

11 vistas

Como parar un script en javascript?

javascript

formulada hace 32 minutos Mohamed El Alami 16

0 votos

0 respuestas

15 vistas

File "main.py", line 7 for {opcion} in lista: ^ IndentationError: expected an indented block (¿Cómo soluciono este error?)

python-2.7

modificada hace 33 minutos afr 169

0 votos

0 respuestas

2 vistas

¿Se puede mostrar proceso de comprobación de un archivo en excel en asp.net core mvc y ajax?

Preguntas populares en la red

When does coming up with an idea constitute sufficient contribution for authorship?

Linux File Manager: Restore previous open session (folders and tabs)

On what did Lego base the appearance of the new Hogwarts minifigs?

It took me a lot of time to make this, pls like. (YouTube Comments #1)

How to approximate rolls for potions of healing using only d6's?

Can you enable Unicorn NewItemsOnly for specific folders?

A starship is travelling at 0.9c and collides with a small rock. Will it leave a clean hole through, or will more happen?

List of numbers giving a particular sum

What is the wife of a henpecked husband called?

Tikzing a circled star

Dilemma of explaining to interviewer that he is the reason for declining second interview

How to acknowledge an embarrassing job interview, now that I work directly with the interviewer?

Why is working on the same position for more than 15 years not a red flag?

I am on the US no-fly list. What can I do in order to be allowed on flights which go through US airspace?

The correct way of using perfect tense

Why is button three on trumpet almost never used alone?

Why does the DC-9-80 have this cusp in its

Imagen tomada de la interfaz de la página de Stack Overflow, Link:
<https://es.stackoverflow.com/>

En la parte izquierda cuenta con un panel que permite navegar por las diferentes vistas (inicio, preguntas, etiquetas, usuarios) y de esta manera moverse por la página. En el parte central cuenta con la zona de preguntas y respuestas, cada publicación de pregunta consta de 3 contadores: votos, respuestas y vistas, también posee un título y una zona de etiquetas que permite hacer referencia al tema de la pregunta. En la parte derecha posee un ranking que redirecciona a las preguntas más populares en la red.

Si se accede a una pregunta aparecerá la siguiente interfaz:



Imagen tomada de la interfaz de las preguntas de la página Stack Overflow, Link:
<https://es.stackoverflow.com/questions/45846/c%C3%B3mo-separar-un-string-en-java-c%C3%B3mo-utilizar-split>

Cada pregunta consta con una descripción, una zona para agregar código, imágenes, o demás elementos que ayuden a comprender la pregunta, una zona de etiquetas y una zona para agregar las diferentes respuestas a la pregunta. En la parte derecha se muestra un listado de preguntas relacionadas a la misma.

Quora

Quora es una red social creada para la distribución masiva de conocimientos, en donde cualquier persona puede realizar una pregunta y cualquiera puede dar una respuesta. (Tomado de <https://es.quora.com/Qu%C3%A9-es-Quora>).

La página de Quora luce de esta manera:

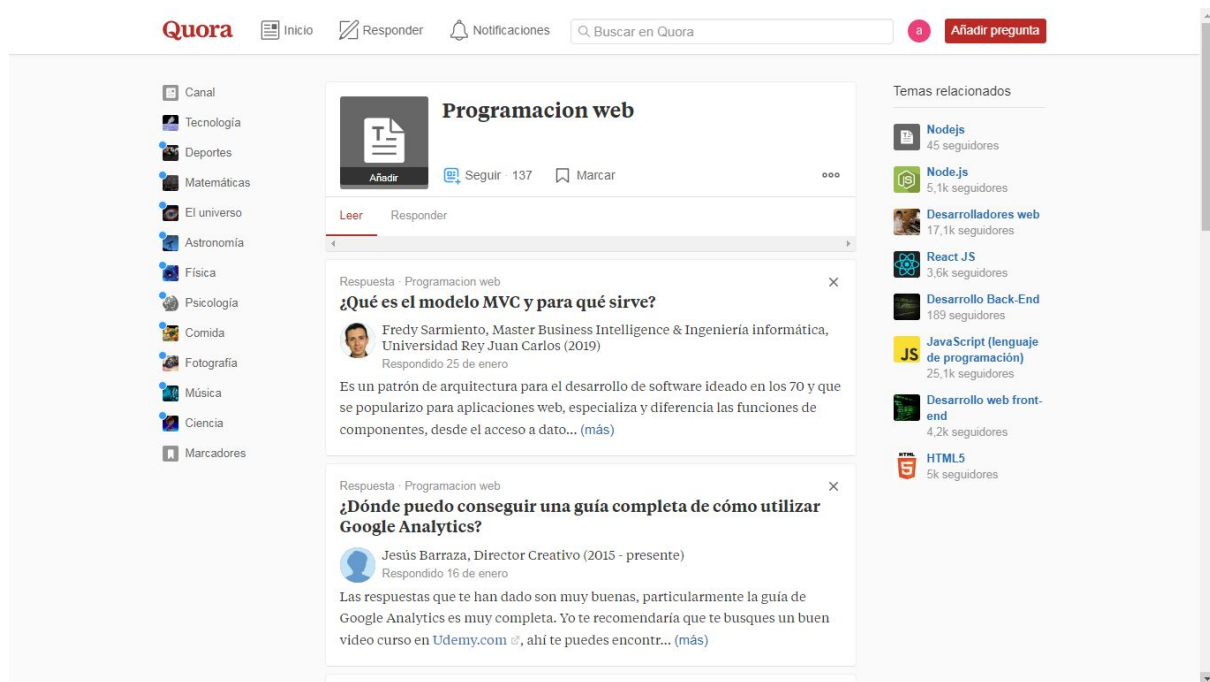


Imagen tomada de la interfaz de la página Quora, Link:
<https://es.quora.com/topic/Programacion-web>

La página posee en su parte izquierda un campo con los canales de interés de cada usuario, esto permite navegar y consultar las preguntas de interés. En su parte media muestra las preguntas más relevantes del canal que seleccionaste, permitiendo ver las respuestas y agregar comentarios a los mismos. En la parte superior posee botones para las acciones principales: ir a inicio, responder, añadir una pregunta y buscar.

Búsqueda de soluciones creativas

Para darle solución al problema se ha decidido dividirlo en cuatro módulos o funcionalidades principales enfocados en un grupo de requerimientos que se relacionan entre sí de la siguiente manera: módulo relacionado con la realización de preguntas, módulo relacionado con la gestión de respuestas, módulo relacionado con la administración de los usuarios y de las preguntas y respuestas realizadas y el módulo de perfiles relacionado con las características de los usuarios y su comportamiento dentro de la aplicación.

Módulo preguntas

En la parte del módulo de preguntas se desarrollará un búsqueda de soluciones a la funcionalidad más relevante y general que es la predicción o sugerencia de etiquetas y *studios* relacionados con la descripción añadida a la pregunta, de la mano con la funcionalidad de encontrar preguntas relacionadas con base a la misma descripción. De esta forma se engloba la problemática a encontrar la manera de sugerir etiquetas, *studios* o preguntas relacionadas de acuerdo a una descripción dada. Para lograrlo se realizó una *lluvia de ideas* de la cual se extraen las siguientes ideas:

1. Implementar una función que tome palabras aleatorias de la descripción escrita, y buscar preguntas con la mayor cantidad de coincidencias

2. Implementar un diccionario que utilice como llaves algunas palabras utilizadas en la descripción, en el cual se almacena un conjunto de preguntas para clasificarlas
3. Implementar un conjunto de diccionarios que almacenen el número de veces que se han encontrado palabras dentro de la descripción, y en base a eso buscar en los diccionarios de otras preguntas una cantidad de palabras similares que sea cercana.
4. Implementar un arreglo por *studio* en el cual se almacenen palabras clave que tengan una relación directa con el *studio* y al escribir o enviar la pregunta, el programa buscará en los arreglos de los *studios*
5. Contar las palabras que se repitan, excluyendo conjunciones entre otras, y realizar una búsqueda dentro de los conjuntos de las palabras más usadas en los arreglos de palabras.
6. Crear un método que ayude a clasificar las preguntas en base a su contenido mediante el análisis de la descripción para proveer la mejor sugerencia
7. Comparar el texto de la descripción con el de otras preguntas y en base a un porcentaje de similitud en palabras se sugiera.
8. Buscar palabras que tengan una importancia dentro de una frase que permita comparar u/o ubicar relaciones con mayor facilidad e implementar sugerencias precisas
9. Implementar un modelo de text analytics que por medio de aprendizaje no supervisado encuentre patrones en las preguntas para buscar relaciones.
10. Utilizar una API que tome parámetros JSON de textos y retorne las palabras claves de dicho texto.

Módulo respuestas

En el módulo de respuestas, se hizo énfasis en aportar y mejorar cada funcionalidad de mayor importancia, para así encontrar el mejor desarrollo de la solución del problema, mediante la lluvia de ideas y así buscar una solución óptima. En esta sección nos enfocaremos en el desarrollo de ideas para el diseño (vista-modelo), dándole mayor importancia a la interacción que se tiene con el usuario. El siguiente listado muestra las diferentes ideas creativas que surgieron para este módulo:

1. Cada usuario tendrá un listado de sus respuestas almacenadas en una estructura de datos, como un Árbol-nArio, donde la raíz es la respuesta principal, y sus descendientes sean las respuestas de la respuesta principal.
2. El usuario tendrá un atributo donde guardara el total de votos positivos, es decir, el atributo almacena la suma de los acumulados de cada pregunta, y así el usuario se clasificará en el ranking de las mejores respuestas.
3. Las respuestas pueden clasificarse como mala, buena y excelente, esto ayudaría a una mejor percepción que tienen los usuarios sobre cada respuesta, y por ende, la clasificación de las respuestas sería mucho más precisa.
4. La clasificación de las respuesta a una pregunta, se visualizará de muy buena a no muy buena, esto quiere decir, que no se tiene en cuenta el tiempo o fecha publicación, se tendrá en cuenta la calidad de la respuesta y por ende se mostrará de mejor a menor calidad. Lo mismo se hará para las respuestas de la respuesta principal.
5. Las respuestas que se consideran de mala calidad, se borran y dan aviso al usuario que le pertenece la respuesta, esto garantiza que el listado de respuesta de una pregunta sea de mejor

calidad, cuando se habla de respuestas malas caen en las que no aportan en la interacción, ni en la solución del problema.

6. La sección de agregar comentario se mantiene siempre activa con su formulario disponible bajo cada respuesta.
7. La sección de agregar comentario a una respuesta se muestra al accionar un elemento que despliegue el formulario para realizar la tarea.
8. Cada respuesta tendrá una barra de llenado, que cambiara de color, dependiendo del nivel de votos positivos que tenga, esta barra funciona con dos colores verde y gris, verde para una alto nivel de votos y gris, con un nivel bastante bajo, esto se hace para una mejor interacción con el plataforma-usuario.
9. El usuario podrá organizar y/u ordenar el listado de respuestas según el criterio de ordenamiento que él desee, los criterios disponibles para el usuario son, antigüedad de la respuesta, nivel de votos positivos y por último, ordena alfabéticamente las respuestas en base al estudio al que pertenece el usuario que respondió.
10. Con base en una opción de búsqueda el usuario podrá buscar respuestas, relacionadas al tema del problema, visualizando así el listado de respuestas acorde con el tema.
11. En el menú principal se asignará un espacio donde el usuario pueda ver el Ranking de usuarios mediante una lista, el criterio de ordenamiento se refleja mediante el promedio de cada usuario, esto se calcula con el número de votos positivos y el número total de respuestas.

Módulo administración

El módulo de administrador, nos permite diferenciar funciones o roles de cada usuario, el administrador se va a encargar de otorgar los derechos de acceso necesarios para cada persona que utilice su cuenta y además podrá realizar un seguimiento del usuario que haya efectuado algún tipo de pregunta. Para poder efectuar todo lo anterior se realizó una lluvia de ideas de las cuales se obtuvo lo siguiente:

Archivar preguntas

1. Administrar el perfil y los derechos de acceso de cada usuario, permitiendo modificar el rol de un usuario cuando sea necesario. Los administradores se pondrán dentro de una estructura de datos para tener más organizado los datos dentro de la aplicación, como por ejemplo:
 - a. Un árbol n-ario en donde las jerarquías se puedan ver en cuanto un administrador colocó a un usuario como administrador, así el que coloca al usuario como administrador sería el papá de este.
 - b. Una pila de datos en donde se pueda ver que administrador fue el primero el ser agregado e ir mirando las jerarquías que tienen por más que tengan los mismos derechos de edición.
 - c. Una lista enlazada sería una estructura que ayude a organizar a los administradores conforme se va agregando al módulo administrativo, así como una lista de tiempo.

- d. Un ArrayList que agregue a los usuarios que accedieron al módulo de administración para tener la información de que usuarios fueron.
- 2. Limitar el acceso, de forma que los usuarios solo puedan encargarse de sus funcionalidades
 - a. Se usará un atributo de tipo boolean el cual se encargue de verificar si es un administrador o un usuario.
 - b. Se podrá buscar en una lista que contenga todos los administradores del sistema diferenciados por Studios.
 - c. Se buscará en la contenedora de los administradores de dicho Studio para poder separar la información dependiendo si son administradores o usuarios.
 - d. Se tendrá una interfaz diferente para que los administradores puedan hacer los cambios que requieran, como eliminar preguntas, además de poder ver información que solo está disponible para los administradores.
- 3. El administrador se debe encargar de la información sobre el entorno, para lograrlo el administrador debe estar en constante monitoreo, para que capte oportunamente cada una de las funcionalidades
 - a. Al administrador se le notificarán las acciones que realicen los usuarios, por medio del correo electrónico ingresado a la hora de registrarse,
 - b. Cualquier consulta realizada por el usuario se almacenará en una lista la cual podrá ser controlada desde el perfil del administrador.
 - c. Habrá una sección de reporte de problemas, el cual el administrador estará en capacidad de acceder a él para poder enviar a los programadores en busca de la mejor solución.
- 4. Archivar preguntas: Se implementará un atributo lógico que permita ocultar la información de la pregunta que permitirá verse como archivada, y el atributo se podrá cambiar con permiso del administrador para poder desarchivar la pregunta.
 - a. Se podrá agregar en una estructura de árbol n-ario en donde la jerarquía será el orden en el que se van archivando las preguntas.
 - b. Se agregaran a una lista enlazada en donde se relacionan por el orden en que se van archivando.
 - c. Se agregaran a un ArrayList que permita tener un atributo como un comentario que permita poner el porque fue archivada esa pregunta.
 - d. Se agrega a una cola, el cual va archivando cada una de estas por orden de llegada.
 - e. Se agrega a una pila el cual se va archivando cada una de estas por orden de llegada, el cual se puede visualizar de forma que la última que llega es la primera que se visualizará
- 5. Eliminar preguntas: Se implementará un atributo lógico que permita ocultar la información de la pregunta que permitirá verse como borrada.
 - a. Se da la posibilidad de que la pregunta sea eliminada lógicamente, esto realizado por medio de un boolean, el cual se añadirá a una lista donde permanecerán durante un determinado tiempo.
 - b. La opción de eliminar se puede visualizar por medio de un combo-box
 - c. Se podrá agregar en una estructura de árbol n-ario en donde la jerarquía será el orden en el que se van eliminando las preguntas.
 - d. Se agregaran a una lista enlazada en donde se relacionan por el orden en que se van eliminando.

- e. Se agregaran a un ArrayList que permita tener un atributo como un comentario que permita poner el porque fue eliminada esa pregunta.
- f. Se agrega a una cola, el cual va eliminando cada una de estas por orden de llegada.
- g. Se agrega a una pila el cual se va eliminando cada una de estas por orden de llegada, el cual se puede visualizar de forma que la última que llega es la primera que se visualizará

Módulo perfiles

En el módulo de perfil de usuario, identificamos todas las necesidades que se deben tener en cuenta acerca de la administración de la sesión de un usuario así como la información que esté puede ver de sí mismo y de otros acorde a las funcionalidades de la aplicación web. Para poder realizar todo lo anteriormente mencionado, se propuso varias ideas de cómo se debe implementar los servicios. Fueron propuestos por lluvia de ideas:

1. Inicio de sesión en la aplicación web. Mediante la cual un usuario, anteriormente registrado, podrá interactuar en la aplicación web.
 - a. Inicio de sesión mediante Google. Mediante un API ofrecido por los servicios de Google, se enlaza esta a la aplicación para que pueda iniciar sesión un usuario anteriormente registrado y vinculado su cuenta de Google.
 - b. Inicio de sesión mediante una base de datos de la empresa Globant. Mediante una base de datos de registro de los usuarios de la compañía, los usuarios podrán iniciar sesión introduciendo sus datos anteriormente registrados en Globant.
 - c. Inicio de sesión por medio de un perfil de facebook. Esta puede ser una segunda opción o alternativa de respaldo para que un usuario de Globant entre a la plataforma. Por lo tanto, la aplicación puede tener varios método de inicio de sesión.
 - d. Inicio de sesión mediante la huella dactilar registrada del usuario en la base de datos de la compañía o aplicación web.
 - e. Inicio de sesión mediante la identificación del rostro del usuario registrado en la base de datos de la compañía o aplicación web.
2. Páginas de perfil de un usuario. Mediante la cual un usuario podrá visualizar toda la información necesaria de sí mismo u otro usuario.
 - a. Implementar página de perfil del usuario como las de facebook donde toda la información básica de un usuario se muestre en una página.
 - b. Implementar página de perfil como la de linkedin donde se muestran los datos necesarios del usuario. Aparte, en otras páginas, se muestran las preguntas realizadas y las respuestas dadas.

- c. Implementar página de perfil como la de StackOverflow donde en la parte superior de la página están ubicadas dos tabs, la primera tab está relacionada con la información del perfil del usuario mientras que la segunda tab muestra las preguntas realizadas como las respuestas realizadas.

3. Notificaciones. Estas informan a los usuarios que han planteado preguntas en la plataforma y otros han respondido a ellas.

- a. Notificar a los usuarios cuando sus preguntas son respondidas mediante su correo o email empresarial.
- b. Notificaciones flotantes cuando un glober responde las preguntas de otro. Estas notificaciones aparecerán en el escritorio durante unos 10 segundos.
- c. Notificaciones en la misma aplicación, es decir, tener un apartado de interactividad donde las notificaciones puedan verse de forma cronológica. Y cuando una o varias de ellas son revisadas eliminarlas del apartado.
- d. Notificaciones mediante el correo empresarial y dentro de la misma plataforma. Al notificarse por ambas partes podemos tener un historial de interactividad, es decir, cuando un usuario revisa las notificaciones desde la aplicación éstas ya no vuelven a aparecer en ella, sin embargo se pueden mirar cuantas veces quiera desde el correo empresarial.

Bibliografía

“API de administración: autorización,” *google.com*, última actualización: 23 de Enero, 2019.[Online]. Disponible en: <https://developers.google.com/analytics/devguides/config/mgmt/v3/authorization?hl=es> [Visitado en Marzo 10,2019]

“Api de Google”, *wikipedia.com*. Publicado el 17 de Junio, 2018.[Online] Disponible en: https://es.wikipedia.org/wiki/API_de_Google [Visitado en Marzo 10, 2018]

A. Sharma. “Authentication Using Google In ASP.NET Core 2.0” *c-sharpcorner.com*, publicado el 12 de Marzo, 2018. [Online]. Disponible en: <https://www.c-sharpcorner.com/article/authentication-using-google-in-asp-net-core-2-0/> [Visitado en Marzo 10,2019]

“Cómo obtener resultados de la búsqueda más rápido” *google.com*, 2019. [Online]. Disponible en: <https://support.google.com/websearch/answer/106230?co=GENIE.Platform%3DAndroid&hl=es-419> [Visitado en Marzo 10,2019]

C. Rojas. “¿Qué es Quora?” *quora.com*, publicado el 6 de Abril, 2017.[Online]. Disponible en: <https://es.quora.com/Qué-es-Quora> [Visitado en Marzo 10, 2019]

D. Pérez. “¿Qué son las bases de datos?” *maestrosdelweb.com*, publicado el 26 de Octubre, 2007. [Online]. Disponible en: <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/> [Visitado en Marzo 10,2019]

“Extracting Keywords From Short Text,” *towardsdatascience.com*. Publicado el 27 de Julio, 2017.[Online]. Disponible en: <https://towardsdatascience.com/extracting-keywords-from-short-text-fce39157166b> [Visitado en Marzo 10,2019]

“Keywords Extraction” *rapidapi.com*, última actualización: 19 de Febrero, 2019. [Online]. Disponible en: <https://rapidapi.com/UnFound/api/keywords-extraction2> [Visitado en Marzo 10,2019]

“More Like This Query,” *elastic.co*, 2019. [Online]. Disponible en: <https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-mlt-query.html> [Visitado en Marzo 10, 2019]

M. Rouse. “Análisis de datos,” *techtarger.com*, última actualización: Noviembre 2017. [Online]. Disponible en: <https://searchdatacenter.techtarger.com/es/definicion/Analisis-de-Datos> [Visitado en Marzo 10,2019]

“Políticas de autocompletar” *google.com*, 2019. [Online] Disponible en:
<https://support.google.com/websearch/answer/7368877> [Visitado en Marzo 10, 2019]

“Tf-idf”, *wikipedia.com*. Publicado el 17 Septiembre, 2018. [Online]. Disponible en:
<https://es.wikipedia.org/wiki/Tf-idf> [Visitado en Marzo 10, 2018]

“Okapi BM25”, *wikipedia.com*. Publicado el 22 de Junio, 2018. [Online]. Disponible en:
https://es.wikipedia.org/wiki/Okapi_BM25 [Visitado en Marzo 10, 2018]

“What are the most popular text similarity algorithms” *www.quora.com*. Publicado el 10 de Diciembre, 2015. [Online]. Disponible en:
<https://www.quora.com/What-are-the-most-popular-text-similarity-algorithms> [Visitado en Marzo 10, 2019]