

Universidad Tecnológica de Bolívar

Facultad de Ingeniería

Programa de Ingeniería de Sistemas

Arquitecturas de software

Profesor

Jairo Enrique Serrano Castañeda

Informe Final de Twitter

Integrantes

Raymundo Diaz

Fabian Hernandez

Carlos Villalobos

Arnulfo Torres

Cartagena, Bolívar

viernes, 29 de noviembre de 2019

Introducción

Las redes sociales representan una gran parte de la vida cotidiana de las personas hoy en día. Algunos las utilizan simplemente para comunicarse con otras personas y ver el estado de sus amigos y familiares, mientras que otros la utilizan como método de mercadeo o venta de productos. En cualquiera de los casos podemos ver que se encuentra una utilidad para calcular estadísticos que nos permitan ver el impacto que tiene la actividad de los usuarios sobre los demás.

Un ejemplo de cómo podría ser útil ver el impacto de tus tweets en Twitter o posts en Facebook es ver a qué tipo de público están atrayendo tus publicaciones y así adaptarte para seguir atrayendo más y más audiencia a tu cuenta. Por esta razón, se decidió crear un microservicio que permita visualizar la información de una cuenta de Twitter, sus últimos 5 tweets y cierta información acerca de los mismos, incluyendo la eficiencia (impacto generado en base a los seguidores del usuario) de cada tweet.

Todo esto se puede realizar utilizando nuevas tecnologías capaces de facilitar el manejo de información que transita actualmente en la nube proporcionando facilidad de uso y accesos como tal por medio de API, las API (Interfaces de programación de aplicaciones) no son más que un conjunto de comandos, funciones y protocolos informáticos que permiten a los desarrolladores crear programas específicos para ciertos sistemas operativos. Las API simplifican en gran medida el trabajo de un creador de programas, ya que no tiene que «escribir» códigos desde cero. Estas permiten al informático usar funciones predefinidas para interactuar con el sistema operativo o con otro programa.

Objetivos

1. General

- a. Desarrollar una aplicación que permita ver estadísticas relacionadas con el impacto de una cuenta de red social en sus seguidores.

2. Específicos

- a. Crear una API que permita visualizar cierta información acerca de una cuenta de Twitter y sus últimas cinco publicaciones.
- b. Conectar la API con los demás microservicios de la aplicación a través del Middleware y hacer que se pueda visualizar con el Dashboard.

Requerimientos

Requerimientos no funcionales

1. Disponibilidad

- **Descripción:** El servicio debe tener una disponibilidad del 99,99% de las veces que se intente acceder a él.
- **Prioridad:** Alta.
- **Efecto colateral:** Moderado consumo energético.

2. Usabilidad

- **Descripción:** El servicio debe contar con manuales de usuario estructurados adecuadamente y debe proporcionar mensajes de error que sean informativos.
- **Prioridad:** Media.
- **Efecto colateral:** Disminución en el tiempo de aprendizaje.

Requerimientos funcionales

1. Envío de datos del usuario como JSON

- **Descripción:** Al realizar una petición a la api solicitando la información de un usuario está devolverá la información de dicho usuario.
- **Entrada:** Nombre de usuario
- **Acción:** Mostrar datos
- **Pre-condición:** Ejecute servicio de twitter
- **Post-condición:** JSON disponible
- **Prioridad:** Alta

2. Envío de publicaciones del usuario como JSON

- **Descripción:** Al realizar una petición a la api solicitando las publicaciones de un usuario está devolverá las publicaciones de dicho usuario.
- **Entrada:** Número de publicaciones(No necesario)(5 por defecto)
- **Acción:** Mostrar datos
- **Pre-condición:** Ejecute servicio de twitter
- **Post-condición:** JSON disponible
- **Prioridad:** Alta

Modelo de datos

user	
ID	string
username	string
fullname	string
followers	string
posts	string

Modelo de datos del usuario

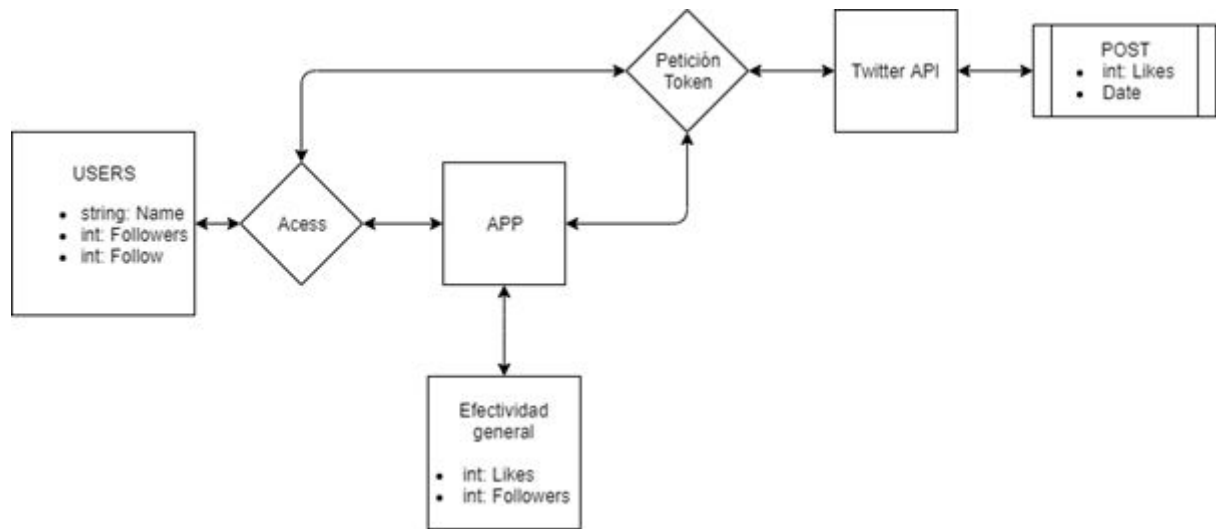
posts	
ID	string
user	string
likes	string
efficiency	string
date	string

Modelo de datos de las publicaciones

Cambios realizados

1. Fue necesario cambiar el tipo de dato de todos los atributos a string para facilitar el manejo de estos por el dashboard.
2. Se cambió el nombre del atributo "id" a "ID" ya que generaba un conflicto al almacenarlo en la colección de la base de datos.
3. Se agregó el atributo "date" en los "posts" para que en el dashboard se mostrará la eficiencia de estos en una línea de tiempo.

Diagrama de funcionalidad API



Metodología Scrum

Se decidió utilizar esta metodología, porque provee un marco de trabajo de procesos ágiles que trabaja con el ciclo de vida iterativo e incremental, donde se va liberando el producto por pares de forma periódica, aplicando las buenas prácticas de trabajo colaborativo (en equipo), facilitando el hallazgo de soluciones óptimas a los problemas que pueden ir surgiendo en el proceso de desarrollo del proyecto.

Con Scrum se realizan entregas regulares y parciales (sprint) del producto final, todas ellas con una prioridad previamente establecida que nace según el beneficio que aportan al cliente, minimizando los riesgos que pueden surgir de desarrollos extremadamente largos. Es por tal motivo, que Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesitan obtener resultados de manera inmediata y donde son fundamentales los siguientes aspectos: la innovación, la productividad, la flexibilidad y la competitividad.

Stakeholder: Jairo serrano (Cliente)

Product Owner: Grupo F Arquitectura software

Scrum Master: Grupo F Arquitectura software

Scrum Team (Equipo de desarrollo): Grupo F Arquitectura software

Proyectos / API TWITTER

Tablero Progreso

AGROUPAR POR: Nada

POR HACER 1	EN CURSO 3	LISTO 8
Presentacion Proyecto. <input checked="" type="checkbox"/> AT-8 + Crear incidencia	Alimentacion de repositorios github <input checked="" type="checkbox"/> AT-6 Generacion de documentacion API y proyecto. <input checked="" type="checkbox"/> AT-7 Pruebas de ejecucion Productivo y resultados. <input checked="" type="checkbox"/> AT-12	Definición de datos en común <input checked="" type="checkbox"/> AT-1 ✓ Definición de requerimientos de desarrollo. <input checked="" type="checkbox"/> AT-10 ✓ Modelo extraccion de API <input checked="" type="checkbox"/> AT-3 ✓ Generación de credenciales y token de acceso API TWITTER <input checked="" type="checkbox"/> AT-4 ✓

Estás en un proyecto de última generación
 Enviar comentarios Más información

Proyectos / API TWITTER

Tablero Progreso

AGROUPAR POR: Nada

POR HACER 1	EN CURSO 3	LISTO 8
		Instalacion Flask y Docker <input checked="" type="checkbox"/> AT-2 ✓ Despliegue servidor y Api de extraccion Funcional (Beta localhost) <input checked="" type="checkbox"/> AT-11 ✓ Pruebas de ejecucion Beta y resultados. <input checked="" type="checkbox"/> AT-9 ✓ Despliegue servidor y Api de extraccion Funcional <input checked="" type="checkbox"/> AT-5 ✓

Estás en un proyecto de última generación
 Enviar comentarios Más información

<https://proyecto211.atlassian.net/jira/software/projects/AT/boards/1> URL DE SEGUIMIENTO.

Herramientas de desarrollo

Docker

Es una herramienta diseñada para facilitar la creación, implementación y ejecución de aplicaciones mediante el uso de contenedores. Los contenedores permiten a un desarrollador empaquetar una aplicación con todas las partes que necesita, como bibliotecas y otras dependencias, y enviarla como un paquete.

Tweepy

Se estableció que todos los microservicios de la aplicación (Reddit, Youtube, Instagram...) deberían utilizar la misma información de las cuentas para calcular la eficiencia de las publicaciones. Por lo tanto, decidimos utilizar librería Tweepy de Python, que nos permite acceder a la API de Twitter y extraer la información necesaria para calcular el impacto de los tweets. Los datos que se utilizaron se dividen en dos; la cuenta que se está consultando y las publicaciones de dicha cuenta. Cada uno de estos datos se despliegan de la siguiente manera:

user

- *ID*: Número entero único con el que se puede identificar la cuenta de Twitter.
- *followers*: Cantidad de seguidores de la cuenta de Twitter.
- *fullname*: Nombre completo de la persona que utiliza la cuenta.
- *post*: Cantidad de posts que ha hecho el usuario de la cuenta.
- *username*: Nombre con el que se identifica la cuenta de Twitter a consultar.

posts

- *ID*: Número de identificación único del tweet.
- *date*: Día, mes, año y hora en que se publicó el tweet.

- *efficiency*: La eficiencia marca el impacto que tuvo un tweet en las personas que lo vieron, es decir, qué tanta atención logró capturar dicha publicación. Para lograr ver la eficiencia de los tweets de una cuenta en Twitter, se utilizó la siguiente fórmula:

$$Eficiencia = \frac{\# \text{ de favoritos del post}}{\# \text{ de followers del usuario}} * 100$$

- *likes*: Número de favoritos que obtuvo la publicación
- *user*: *username* del usuario que hizo la publicación.

Cabe recalcar que para hacer uso de la API de Twitter es necesario tener una credencial de desarrollador. Esta credencial se puede generar fácilmente a través de la página <https://developer.twitter.com/>. Una vez se haya creado una aplicación y generado las llaves de autenticación, estas pueden ser utilizadas para usar el microservicio sin problema.

Flask

Para el desarrollo de esta aplicación capaz de extraer información específica y realizar cálculos u operaciones específicas como la eficiencia de cada post que realice un usuario fue necesaria la utilización de un framework que permita una usabilidad y adaptación de manera sencilla y eficaz a las necesidades planteadas para estas tareas, capaz de permitir un desarrollo claro y ordenado de los requerimientos y procesos a implementar, para el presente caso hemos optado por la utilización del micro frameWork FLASK para nuestro lenguaje de desarrollo a utilizar Python que posee como objetivo mantener el núcleo simple pero extensible.

Flask no tomará muchas decisiones por usted, como qué base de datos usar. Esas decisiones que toma, como qué motor de plantillas usar, son fáciles de cambiar. Todo lo demás depende de usted, para que Flask pueda ser todo lo que necesita y nada que no necesite.

Condiciones de uso.

Para la utilización de cualquier clase de api que se quiera consumir para un proyecto o aplicación como tal se deben tener en cuentas algunas cláusulas o condiciones establecidas por parte de la entidad desarrolladora de dicha API, en este caso twitter tiene ciertas condiciones establecidas para la API utilizada en este proyecto capaz de extraer información necesaria para nuestro proyecto de medición de eficiencia por las publicaciones que puede realizar un usuario en específico, a continuación enunciamos algunas condiciones que están establecidas por parte del creador de la API (Twitter) a utilizar:

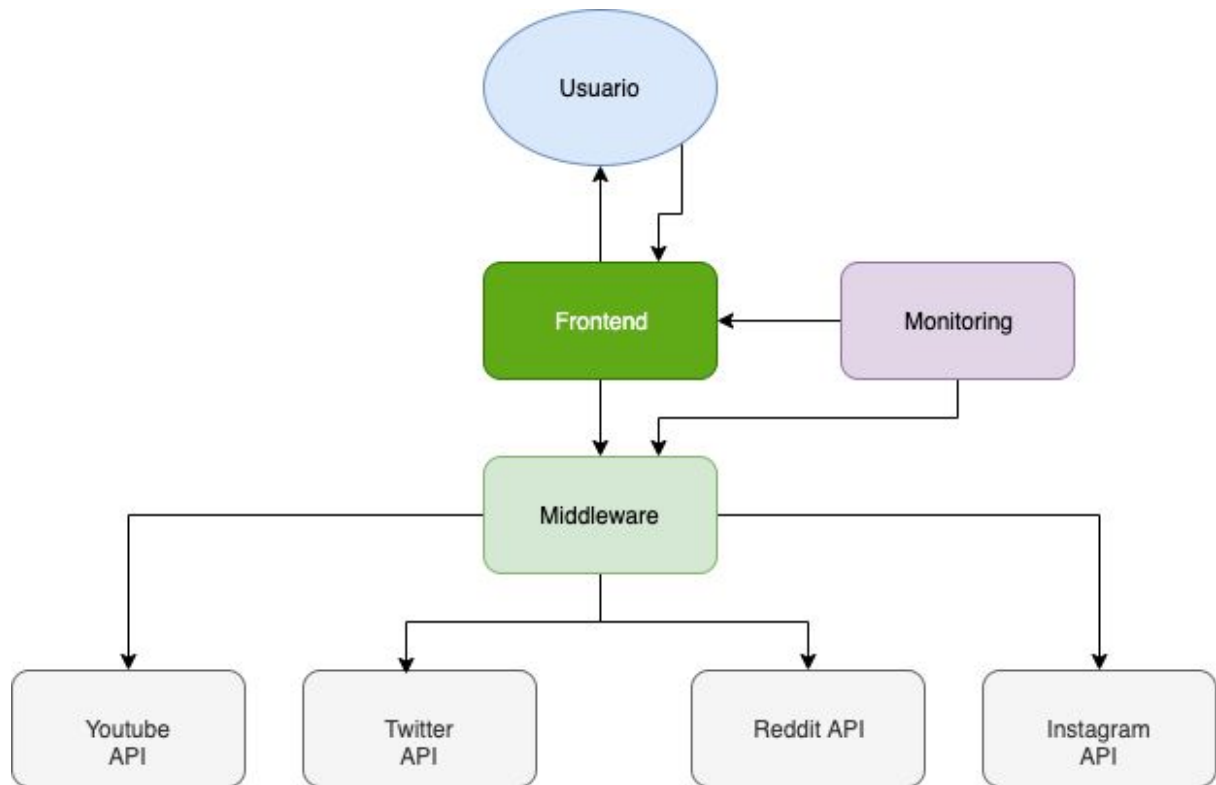
Puedes:

- Desarrollar aplicaciones que transmitan de forma automática información en los Tweets, mientras no incumplas las normas de Tweets automatizados.
- Publicar campañas de creatividades que responden automáticamente a los usuarios que interactúan con tu contenido.
- Desarrollar soluciones que responden a los usuarios de forma automática en los Mensajes Directos.
- Probar cosas nuevas que ayudan a la gente (y que cumplen con nuestras reglas).
- Asegurarte de que tu aplicación ofrezca una buena experiencia de usuario y de que tenga un buen rendimiento, y constata que siga siendo así a lo largo del tiempo.

No puedes:

- Incumplir estas u otras políticas. Ten en cuenta especialmente nuestras reglas sobre las prácticas abusivas y la privacidad del usuario.
- Abusar de la API de Twitter o intentar eludir los límites de velocidad.
- Enviar spam o importunar a los usuarios, o bien enviarles cualquier tipo de mensajes no solicitados.

Diagrama de diseño conjunto de APIs



Avances logrados

1. Se cumplió con los objetivos específicos propuestos.
2. El servicio cumple con los requerimientos establecidos.
3. Se logró desplegar exitosamente el servicio en el servidor.
4. Se probó el servicio en conjunto con toda la aplicación sin ningún inconveniente.
5. El servicio se comunicó exitosamente con el middleware y se logró almacenar la información en la colección.
6. Se realizaron pruebas de comunicación sin problemas con el dashboard, con algunos inconvenientes en despliegue de la información.

Conclusión

La intención principal de este proyecto es poner en práctica la diversidad de conceptos que se vieron durante el semestre en la asignatura de Ing. de Software II de la Universidad Tecnológica de Bolívar, conceptos como Docker, micro frameworks (Flask), Api, micro servicios, patrones de diseños, entre otros más, este proyecto nace de la necesidad de emplear todos los conocimientos adquiridos de manera grupal en donde se desarrollen servicios capaces de interactuar entre ellos y de acoplamiento rápido y efectivo con la finalidad de mostrar o manejar información de manera conjunta y específica, buscar la manera de trabajo en equipo para nuevos proyectos o experiencias en el ámbito de desarrollo tecnológico que encontraremos en la vida laboral de un Ing. en sistemas, aplicando metodologías de desarrollo como la Scrum que permite un desarrollo ágil, organizados y segmentado por etapa de desarrollo o producción, al concluir este trabajo nos damos cuenta que nuestro proceso de formación está basado en un continuo cambio, de conocimiento para el mejoramiento de nuestra labores, conocimiento y experiencia (que aplicamos en base a un estudio que se ha adquirido para mejorar día a día) y con el objetivo de llegar a ser un buen Ing. en sistemas, con las competencias que la materia nos ha suministrado; con responsabilidad, siempre a la expectativa del cambio, para mejoras de un mejor servicio a una sociedad o comunidad ávida de conocimiento y con deseos de crecimiento incesante en cualquier ámbito posible, puntualmente el trabajado en este proyecto, el ambiente tecnológico.

Referencias

- [Flask information](#)
- [API conceptos basicos](#)
- [Tweepy Org](#)
- [Twitter Developers](#)
- [Python Org](#)
- [IngenieriaDeSistemasUTB/ArcSoft2p2019](#) (GitHub Project)
- [Docker](#)