

## **INFORME DE LABORATORIO**

### **Tendencias Disruptivas en TI**

#### **Participante**

Ing. José Hernando Benavides

CC. No. 12.996.632

Ing. Esp. J. Javier Vaquiro

CC. # 14.885.455

#### **Tutor**

Roberto Mauricio Cárdenas

Universidad Nacional Abierta y a Distancia - UNAD

Tendencias Disruptivas en TI

Maestría en Gestión de TI

2016

## Contenido

1. Introducción.....	4
2. Investigar.....	5
a) ¿Qué es Git? .....	5
b) ¿Qué es Github?, .....	5
3. Explicar los siguientes comandos en sus propias palabras.....	6
3.1. Git clone: .....	6
3.2. Git pull: .....	6
3.3. Git push: .....	6
3.4. Git checkout: .....	7
3.5. Git add: .....	7
3.6. Git pull rebase: .....	7
3.7. Git stash:.....	7
3.8. Git stash pop:.....	8
3.9. Git reset: .....	8
3.10. Git cherry – pick:.....	8
4. Investigar.....	9
a) ¿Qué significan las siglas IaaS, PaaS, y SaaS? .....	9
¿Qué es Heroku? .....	11
5. Registrarse en la plataforma de Heroku y descargar Heroku Toolbet para desarrollar aplicaciones.....	11
¿Qué es Node.js?, ¿Cuál es su importancia hoy?, ¿Por qué muchas empresas en mundo usan Node.js? .....	12
¿Qué es EXPRESSJS? De otros ejemplos.....	13
¿Qué son los WebSockets?, ¿Que es Socket.IO? .....	14
8.1. WebSocket. ....	14
Socket.IO .....	14
Investigar.....	14
a) ¿Qué es .....	14
Streaming, aliado para la comunicación digital y la publicidad el streaming permite la posibilidad de disfrutar de contenidos multimedia sin necesidad de descargar previamente el archivo completo en un dispositivo. ....	14
¿Cómo podría una empresa colombiana usar el Twitter Streaming API?.....	15
Diseñe un proyecto que use el Twitter API, y que añada valor al negocio de una empresa en el contexto colombiano. ....	15
Registrarse para crear una aplicación que use el Twitter Streaming API. Generar las diferentes claves de AUTH2 para poder usar el API. ....	15

12. Clone el proyecto <a href="https://github.com/UNADTDTI/laboratorio-practico.git">https://github.com/UNADTDTI/laboratorio-practico.git</a> , y siga las instrucciones en el archivo README. ....	16
13. Conclusiones. ....	17
14. Referencias Bibliográficas ....	18

## 1. Introducción.

Las plataformas de redes han existido desde los años sesenta, iniciándose con conexiones de área local (LAN), evolucionando hacia redes interconectadas en áreas amplias (WAN). Todo este ecosistema ha permitido definir servicios independientes de ubicación y tamaño, teniendo sitios web que aparentemente responden a una sola dirección IP, pero que internamente involucran a una red de nodos que actúan como un sistema dedicado. Ésta es la nube, un ecosistema que permite desplegar servicios sobre redes en base a infraestructuras complejas y escalables, tales como Google o Amazon.

El progreso de las redes de computadoras en los últimos años ha generado nuevos modelos de negocio basados en servicios, tales servicios han también evolucionado al punto de convertirse en infraestructuras completas y confiables con un alto grado de disponibilidad.

## 2. Investigar.

a) **¿Qué es Git?**, ¿Que lo hace diferente a otros sistemas de versionamiento? Git, es un software de control de versiones diseñado por Linus Torvalds. La pregunta es ¿qué es control de versiones? Pues bien, se define como control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo es decir a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración, y para los que aún no les queda claro del todo, control de versiones es lo que se hace al momento de estar desarrollando un software o una página web. Exactamente es eso que haces cuando subes y actualizas tu código en la nube, o le añades alguna parte o simplemente le editas cosas que no funcionan como deberían o al menos no como tú esperarías.

Algunas de las características más importantes de Git son:

- Rapidez en la gestión de ramas, debido a que Git nos dice que un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente.
- Gestión distribuida; Los cambios se importan como ramas adicionales y pueden ser fusionados de la misma manera como se hace en la rama local.
- Gestión eficiente de proyectos grandes.
- Realmacenamiento periódico en paquetes.

b) **¿Qué es Github?**, ¿Por qué es importante para nosotros en este curso? GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

Es importante para nosotros en este curso por varias razones:

- Esa facilidad para compartir código del repositorio alojado en la nube con Gitgub y la misma sencillez que nos ofrece el sistema de control de versiones Git para trabajar, ha permitido que muchos proyectos Open Source se hayan pasado a Github como repositorio y a partir de ahí hayan comenzado a recibir muchas más contribuciones en su código.

- ▶ Quizás nos preguntemos ¿cómo obtienen dinero en Github si alojar proyectos en sus repositorios es gratis? Realmente solo es gratuito alojar proyectos públicos, de código abierto. El servicio también permite alojar proyectos privados y para ello hay que pagar por una cuenta comercial o un plan de hosting que no es gratuito. Existen planes iniciales, para alojar hasta cinco proyectos privados a partir de 7 dólares por mes, lo que resulta bastante barato. Ese mismo plan es gratuito para los estudiantes universitarios. Github además se ha convertido en una herramienta para los reclutadores de empleados, que revisan nuestros repositorios de Github para saber en qué proyectos contribuimos y qué aportaciones hemos realizado. Por ello, hoy resulta importante para los programadores no solo estar en Github sino además mantener un perfil activo.

### **3. Explicar los siguientes comandos en sus propias palabras.**

#### **3.1. Git clone:**

Clones un repositorio en un directorio recién creado, crea ramas de seguimiento remoto para cada rama en el repositorio clonado (visible usando `git branch -r`), y crea y comprueba una rama de inicial que se bifurca de la rama actualmente activo del repositorio clonado. Después de que el clon, una llanura `git fetch` sin argumentos actualizará todas las ramas de seguimiento a distancia, y un `git pull` sin argumentos será además fusionar la rama master a distancia en la actual rama principal, en su caso (esto no es cierto cuando `--single-rama` es dado). Esta configuración predeterminada se logra mediante la creación de referencias a los directores de las sucursales remotas bajo `refs/remotes/originy` inicializando `remote.origin.url` `remote.origin.fetch` variables de configuración.

#### **3.2. Git pull:**

Este comando nos baja los cambios que haya trabajado otra persona en el mismo repositorio, es recomendable siempre que trabajemos un repositorio que no hayamos iniciado nosotros hacer siempre un `git pull` primero antes de comenzar a trabajar en los archivos.

#### **3.3. Git push:**

Baja los cambio de la rama determinada y la coloca en una rama "espejo" que simplemente es una clase de rama "escondida" en la cual tu puedes mirar los cambios de dicha rama, para posterior hacer merge con tu rama local.

### 3.4. Git checkout:

Actualizaciones de los archivos en el directorio de trabajo para que coincida con la versión en el índice o el árbol especificado. Si no se dan caminos, git checkout también actualizará CABEZA para establecer la rama especificada como la rama actual.

### 3.5. Git add:

Este comando actualiza el índice mediante el contenido actual que se encuentra en el directorio de trabajo, para preparar el contenido organizado para el próximo commit. Normalmente se añade el contenido actual de los caminos existentes en su conjunto, pero con algunas opciones que también se puede utilizar para agregar contenido con sólo una parte de los cambios realizados en los archivos del árbol de trabajo aplicado, o eliminar caminos que no existen en el árbol de trabajo nunca más.

Git checkout – b “nombre”:

Crea una rama y comprueba hacia fuera.

Git commit – am”primer commit”:

Usando el parámetro –a SIGNIFICA Que Va un buscas? Todos Los Archivos Modificados al cometió.

### 3.6. Git pull rebase:

Incorpora cambios con respecto a un repositorio remoto en la rama actual. Más precisamente, git pull ejecuta git fetch con los parámetros dados y llama git merge para combinar los directores de las sucursales recuperados en la rama actual. Con --rebase, se ejecuta git rebase en lugar de git merge .

### 3.7. Git stash:

Utilizar git stash cuando se desea registrar el estado actual del directorio de trabajo y el índice, pero quiero volver a un directorio de trabajo limpio. El comando guarda sus modificaciones locales de distancia y revierte el directorio de trabajo para que coincida con la HEAD confirmación.

### 3.8. Git stash pop:

Aplica cambios de un guardado y lo retira inmediatamente de la pila

### 3.9. Git reset:

Cuando se invoca con una ruta de archivo, git resetse actualiza la instantánea en escena para que coincida con la versión de la confirmación especificada.

### 3.10. Git cherry – pick:

Teniendo en cuenta una o más confirmaciones existentes, se aplican el cambio de cada una de ellas introduce, la grabación de un nuevo commit para cada uno. Esto requiere que su árbol de trabajo para ser limpio (no hay modificaciones de la cabeza se comprometen).





## Computación en la nube

### 4. Investigar.

- a) ¿Qué significan las siglas IaaS, PaaS, y SaaS? Explique las diferencias con ejemplos prácticos.

#### **IaaS:**

El concepto de Infraestructura como Servicio (IaaS, Infrastructure as a Service) es uno de los tres modelos fundamentales en el campo del cloud computing, junto con el de Plataforma como Servicio (PaaS, Platform as a Service) y el de Software como Servicio (SaaS, Software as a Service). Al igual que todos los servicios cloud, IaaS proporciona acceso a recursos informáticos situados en un entorno virtualizado, la "nube" (cloud), a través de una conexión pública, que suele ser internet. En el caso de IaaS, los recursos informáticos ofrecidos consisten, en particular, en hardware virtualizado, o, en otras palabras, infraestructura de procesamiento. La definición de IaaS abarca aspectos como el espacio en servidores virtuales, conexiones de red, ancho de banda, direcciones IP y balanceadores de carga. Físicamente, el repertorio de recursos de hardware disponibles procede de multitud de servidores y redes, generalmente distribuidos entre numerosos centros de datos, de cuyo mantenimiento se encarga el proveedor del servicio cloud. El cliente, por su parte, obtiene acceso a los componentes virtualizados para construir con ellos su propia plataforma informática.

#### **PaaS:**

El concepto de Plataforma como Servicio (PaaS, Platform as a Service) es una categoría de servicios cloud que proporciona una plataforma y un entorno que permiten a los desarrolladores crear aplicaciones y servicios que funcionen a través de internet. Los servicios PaaS se alojan en la nube, y los usuarios pueden acceder a ellos simplemente a través de su navegador web.

El modelo PaaS permite a los usuarios crear aplicaciones de software utilizando herramientas suministradas por el proveedor. Los servicios PaaS pueden consistir en funcionalidades preconfiguradas a las que los clientes puedan suscribirse, eligiendo las funciones que deseen incluir para resolver sus necesidades y descartando aquellas que no necesiten. Así, los paquetes pueden variar desde un sencillo entorno que se maneje con el

ratón y no requiera ningún tipo de conocimiento o instalación especial por el lado del usuario, hasta el suministro de opciones de infraestructura para desarrollo avanzado.

Estas son algunas de las funcionalidades que pueden incluirse dentro de una propuesta de PaaS:

1. Sistema operativo
2. Entorno de scripting de servidor
3. Sistema de gestión de base de datos
4. Software de servidor
5. Soporte técnico
6. Almacenamiento
7. Acceso a la red
8. Herramientas de diseño y desarrollo
9. Hosting

### **SaaS:**

Con el concepto de Software como Servicio (SaaS, Software as a Service) se describe cualquier servicio cloud en el que los consumidores puedan acceder a aplicaciones de software a través de internet. Esas aplicaciones están alojadas "en la nube" y pueden utilizarse para una amplia variedad de tareas, tanto para particulares como para organizaciones. Google, Twitter, Facebook y Flickr son ejemplos de SaaS, en los cuales los usuarios pueden acceder a los servicios a través de cualquier dispositivo que pueda conectarse a internet. Los usuarios empresariales pueden utilizar aplicaciones para resolver necesidades muy diversas, desde la contabilidad y la facturación hasta el seguimiento de ventas, planificación, control de rendimiento y comunicaciones (por ejemplo, el correo web y la mensajería instantánea).

El modelo SaaS se conoce también a veces como "software a demanda", y la forma de utilizarlo se parece más a alquilar el software que a comprarlo. Con las aplicaciones tradicionales, el software se compra al principio como un paquete, y una vez adquirido se instala en el ordenador del usuario. La licencia del software puede también establecer limitaciones en cuanto al número de usuarios y/o dispositivos en los cuales puede instalarse. Por el contrario, los usuarios del Software como Servicio se suscriben al software, en lugar de comprarlo, generalmente por períodos mensuales. Las aplicaciones se compran y utilizan a través de internet, y los archivos se guardan en la nube, no en el ordenador del usuario.

## ¿Qué es Heroku?

[Heroku](#) es una plataforma en la nube que permite a las empresas a construir, entregar, supervisar y aplicaciones escala; es la manera más rápida para ir de la idea al URL, sin pasar por todos esos problemas de infraestructura.

Heroku desarrollada desde junio de 2007, con el objetivo de soportar el lenguaje de programación Ruby, pero después se ha extendido el soporte a Java, Node.js, Scala, Clojure y Python y PHP. La base del sistema operativo es Debian.

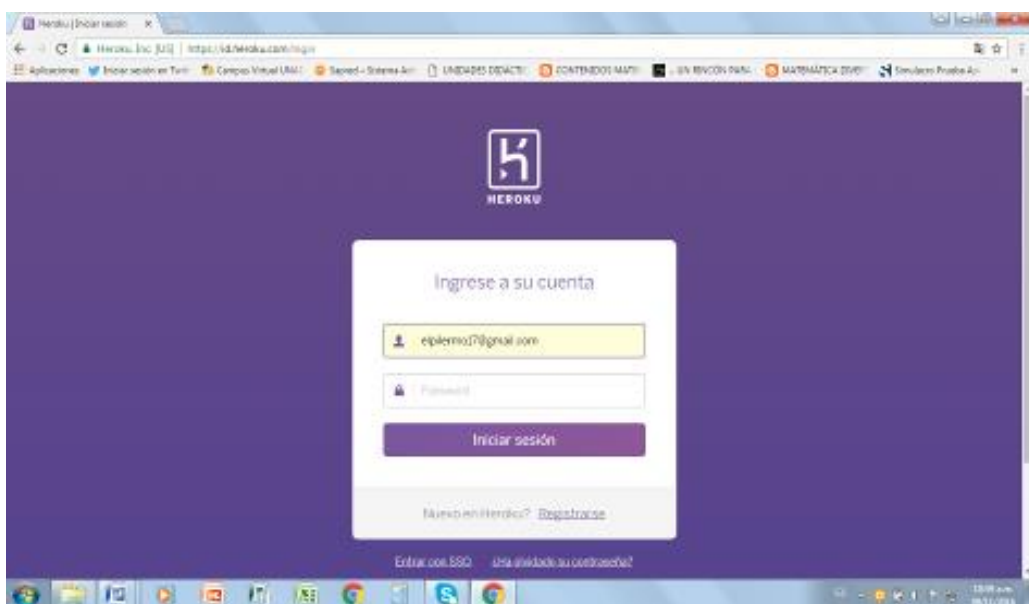
**5. Registrarse en la plataforma de Heroku y descargar [Heroku Toolbet](#) para desarrollar aplicaciones.**

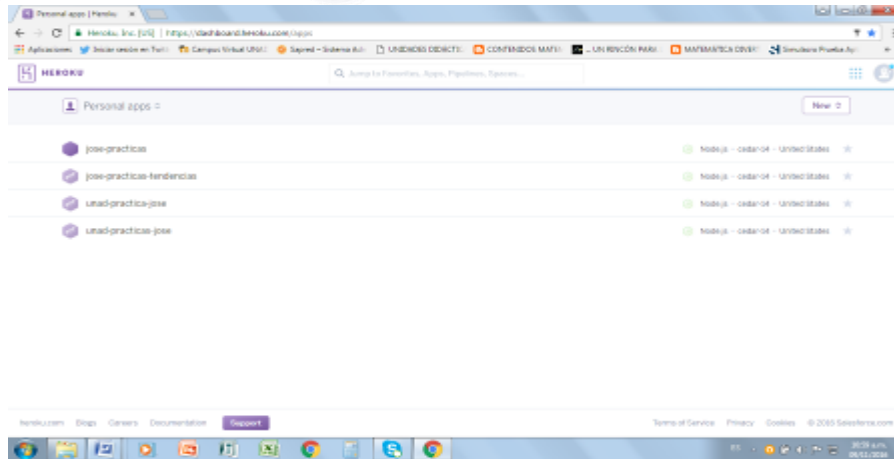


```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\trabajo>heroku login
Enter your Heroku credentials.
Email: elpilermol7@gmail.com
Password (typing will be hidden):
Logged in as elpilermol7@gmail.com

C:\Users\trabajo>
```





## Herramientas de Software

**¿Qué es Node.js?, ¿Cuál es su importancia hoy?, ¿Por qué muchas empresas en mundo usan Node.js?**

**Qué es Node.js:**

Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como, por ejemplo, servidores web. Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla.

**Cuál es su importancia hoy:**

La importancia que presenta Node.js es por varias razones, pero una de las más importantes es el gran número de frameworks y herramientas para pruebas realizados en torno a node.js.

Uno de nuestros favoritos, y sobre el que vamos a hablar mucho próximamente es Nightwatch.js, que permite realizar pruebas automáticas de navegación web de manera sencilla, utilizando node y Selenium server.

## Por qué muchas empresas en mundo usan Node.js:

Empresas como Uber, eBay e incluso el “New York Times” utilizan Node.JS dentro de sus aplicaciones, considerando las cantidades enormes de recurso que se mueven ahí, que es muy poco probable que pongan sus modelos de negocio en tecnologías inestables.

### **Esto genera confianza.**

Y la confianza en una tecnología siempre formará comunidad, la cual ésta aportará mejoras continuas, volviéndola más robusta, segura, ágil y con alto retorno de inversión para nuestros proyectos.

Resultado de todo esto ha generado frameworks (entornos de desarrollo) muy interesantes que te permiten realizar aplicaciones elegantes y escalables.

## ¿Qué es EXPRESSJS? De otros ejemplos

Express.js , o simplemente expreso , es un framework de aplicaciones web para Node.js , liberado como software libre y de código abierto bajo la licencia MIT. Está diseñado para la construcción de aplicaciones web y API. Es el marco de servidor estándar de facto para Node.js. El autor original, TJ Holowaychuk, lo describió como un Sinatra servidor, inspirada, lo que significa que es relativamente mínima con muchas características disponibles como plugins. Express es la parte de back-end de la pila MEDIA, junto con MongoDB base de datos y AngularJS marco frontend.

### **7.1. Ejemplos:**

#### **Meteor , o MeteorJS**

Meteor permite la creación rápida de prototipos y produce multiplataforma ( Android , iOS , web de código). Se integra con MongoDB y utiliza el Protocolo de datos distribuidos y un modelo de publicación-suscripción para propagar automáticamente los cambios de datos a los clientes sin que el desarrollador escribir ningún código de sincronización.

#### **Socket.IO**

Es un JavaScript biblioteca en tiempo real de aplicaciones web. Se permite en tiempo real, la comunicación bidireccional entre los clientes y los servidores web.

## ¿Qué son los WebSockets?, ¿Que es Socket.IO?

### 8.1. WebSocket.

Es una tecnología que proporciona un canal de comunicación bidireccional y [full duplex](#) sobre un único [socket TCP](#). Está diseñada para ser implementada en [navegadores](#) y [servidores web](#), pero puede utilizarse por cualquier aplicación cliente/servidor.

#### Socket.IO

Es una librería en JavaScript para Node.js que permite una comunicación bidireccional en tiempo real entre cliente y servidor. Para ello se basa principalmente en Websocket pero también puede usar otras alternativas como sockets de Adobe Flash, JSONP polling o long polling en AJAX, seleccionando la mejor alternativa para el cliente justo en tiempo de ejecución.

### Social media

#### Investigar

- a) **¿Qué es Twitter Streaming API?** Dar un ejemplo de su uso en productos comerciales o por parte de alguna empresa.

Significa Interfaz de programación de aplicaciones. Es una herramienta que permite la interacción con los programas informáticos y servicios web fáciles. Muchos servicios web proporciona APIs para los desarrolladores para interactuar con sus servicios y para acceder a los datos en forma programática. Para este tutorial, vamos a utilizar la API de Twitter Streaming para descargar los tweets relacionados con 3 palabras clave: "python", "javascript" y "ruby".

#### Ejemplos:

Streaming, aliado para la comunicación digital y la publicidad el streaming permite la posibilidad de disfrutar de contenidos multimedia sin necesidad de descargar previamente el archivo completo en un dispositivo.

Transmite tu evento en vivo a cualquier lugar del mundo, aumenta la audiencia y maximiza su impacto, gracias a la integración con las redes sociales. Mejora la experiencia de los



usuarios de tus eventos, ofreciendo vídeo en alta calidad, accesible desde múltiples dispositivos.

### ¿Cómo podría una empresa colombiana usar el Twitter Streaming API?

En Streaming Colombia somos especialistas y profesionales como proveedores de plataformas para Streaming, WebTV y Radio Online. Transmitimos eventos en directo por internet, con producción audiovisual al mismo nivel que la televisión y con equipos de última tecnología. Nos hemos aliado con las compañías tecnológicas más importantes del sector en América y Europa para ofrecer servicios de streaming a números ilimitados de audiencias en cualquier lugar del mundo y a múltiples dispositivos.

### Diseñe un proyecto que use el Twitter API, y que añada valor al negocio de una empresa en el contexto colombiano.

Esta pregunta deberá contener una motivación desde el punto de vista de negocio y un diseño holístico de la solución de TI.

Ver Link del sitio

<http://tdisruptivas.blogspot.com.co/>

### Registrarse para crear una aplicación que use el Twitter Streaming API. Generar las diferentes claves de AUTH2 para poder usar el API.





12. Clone el proyecto <https://github.com/UNADTDI/laboratorio-practico.git>, y siga las instrucciones en el archivo README.

Después de terminar las modificaciones, despliegue la aplicación web en Heroku, e incluya la URL de la Aplicación corriendo.



<https://jose-practicas-tendencias.herokuapp.com/>



### 13. Conclusiones.

Es importante en el desarrollo y la depuración de una aplicación que el entorno de desarrollo local se ejecuta de la misma manera que los entornos remotos. Esto asegura que las incompatibilidades y difícil de encontrar errores son capturados antes de desplegar a la producción y trata la aplicación como una unidad integral en lugar de una serie de comandos individuales que trabajan de forma independiente.

Es una herramienta de línea de comandos para ejecutar aplicaciones de copia. Se instala automáticamente por el Heroku Toolbelt. Las aplicaciones Heroku están preparadas para poder tener una muy alta escalabilidad. Esto se consigue utilizando procesos, que son instancias independientes que Heroku puede levantar y bajar en cualquier momento (en función de la carga de trabajo). Cada instancia contiene una copia del código fuente de la aplicación. Es la plataforma Heroku la que se ocupa de colocar un servidor web delante, y realiza las comunicaciones SSL, etc. Los procesos pueden tener una vida muy efímera (por ejemplo, para atender tan solo una petición) con lo cual no es recomendable almacenar información en memoria, ni tampoco escribir ningún fichero en disco (en su lugar, se recomienda utilizar servicios de Amazon).



## 14. Referencias Bibliográficas

<https://es.wikipedia.org/wiki/Heroku>

<https://devcenter.heroku.com/categories/command-line>

<https://devcenter.heroku.com/articles/multiple-environments>

<http://stackoverflow.com/questions/4413681/git-push-vs-git-push-heroku-master>

<https://geekytheory.com/como-usar-la-api-de-twitter-en-php/>

<http://www.programmableweb.com/api/twitter>