

TECNOLOGICO DE COSTA RICA

TI-3404 - Lenguajes de Programación

Prof.: Andréi Fuentes Leiva



“Buscador de hashtags de Instagram y Twitter”

Documentación Técnica

Víctor Vargas Ramírez

José Pablo Matamoros Moya

Tabla de contenidos

Descripción del problema	3
Diseño del programa	3
Acerca del programa	4
API de Twitter	4
API de Instagram	5
Forma de obtener el contenido de los APIs	5
Twitter	5
Instagram	6
Forma de almacenar la información	6
Lenguajes utilizados	7
Ruby on Rails	7
→ SINATRA (Framework)	7
CSS3 (Cascading Style Sheet)	7
JavaScript	7
→ jQuery (Biblioteca de JavaScript)	7
Clases y Métodos utilizados	8
Análisis de resultados	9
Objetivos Alcanzados	9
Objetivos no Alcanzados	9

Descripción del problema


El propósito de este programa será el análisis de hashtags, se buscará un contenido explícito por el usuario, el cual indicará un criterio de búsqueda que será posteriormente utilizado para obtener los resultados de dicha búsqueda en las redes sociales Twitter y en Instagram con base en los hashtags y contenido de las publicaciones de los usuarios de las redes sociales relacionadas al criterio.

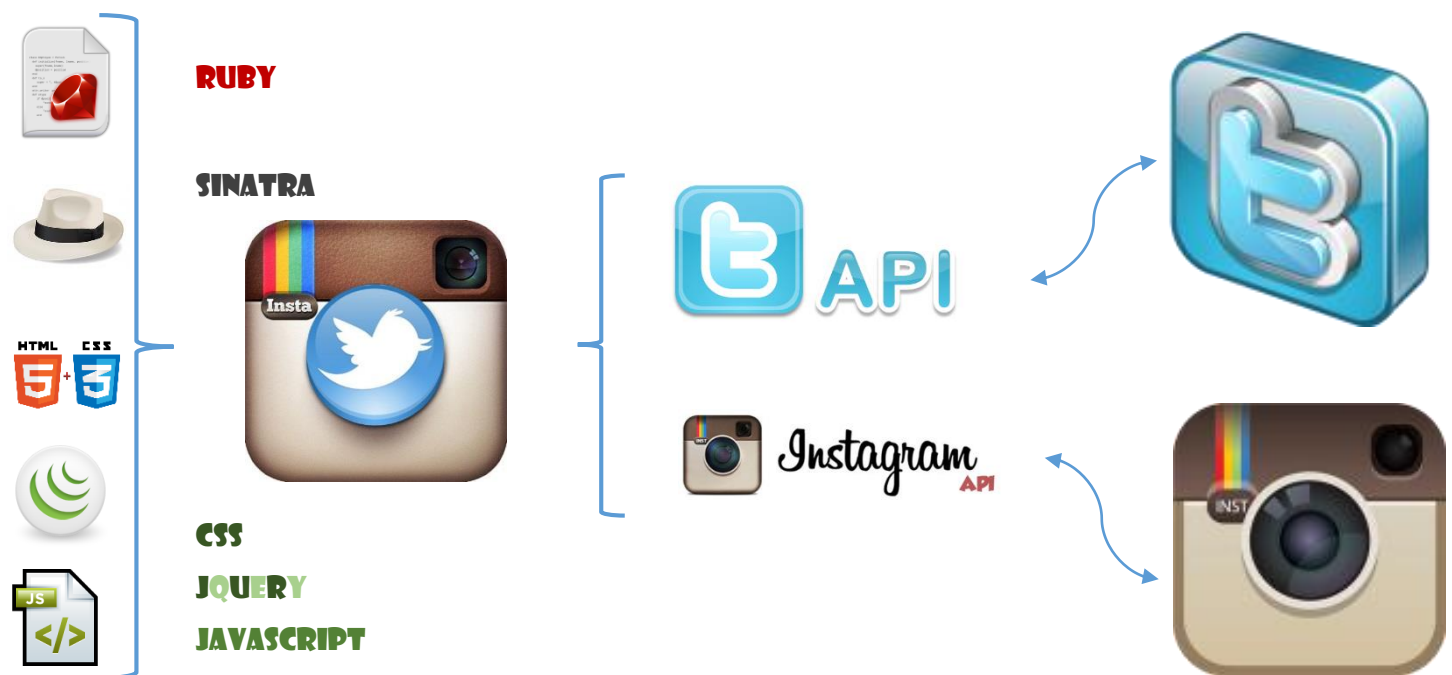
El programa indicará los resultados de la búsqueda y devolverá los tweets y las imágenes relacionadas a esta. El usuario tendrá la opción de indicar la cantidad de resultados que desea ver (tweets que cumplen criterio) y estos se mostrarán en el resultado de la búsqueda.

El programa se conectará el API¹ de búsqueda de Twitter y de Instagram para poder llevar a cabo dichas consultas y obtener los resultados. Cada resultado deberá contener información básica ya sea del tweet o de la foto, los cuales deben ser:

Tweet	Foto
Texto (Tweet completo publicado)	Fotografía respectiva
Timestamp (Hora y fecha de la publicación)	Timestamp
Usuario (Usuario que publicó el tweet)	Usuario
Foto usuario (Foto del usuario en Twitter)	

Diseño del programa

El siguiente esquema muestra una idea resumida de la estructuración del programa . El mismo está construido con distintos lenguajes de programación y se comunican con los servidores mediante el manejo de API's como podemos verlo a continuación:



¹ Una **API** es una interfaz de programación de aplicaciones. Son publicadas por los constructores de software para permitir acceso a características de bajo nivel o propietarias, detallando solamente la forma en que cada rutina debe ser llevada a cabo y la funcionalidad que brinda, sin otorgar información acerca de cómo se lleva a cabo la tarea. Son utilizadas por los programadores para construir sus aplicaciones sin necesidad de volver a programar funciones ya hechas por otros, reutilizando código que se sabe que está probado y que funciona correctamente.

Acerca del programa

Lo primero es comprender que son las hashtags, estos son un grupo de metadatos los cuales contienen información acerca de la publicación, son datos acerca de otros datos, lo cual sirve para identificar y organizar la información utilizando ciertos valores o palabras que son relevantes acerca de estos, términos clave que describen acerca del tema que se trata en la publicación o imagen, estas permiten obtener publicación que giren en torno a un tema en específico y filtra la información de manera que podamos utilizar solo la que queremos. Estas palabras claves son ampliamente utilizadas en redes sociales para crear conversaciones que giren alrededor de un tema determinado y podamos crear grupos de conversaciones o publicaciones que hablan acerca del mismo tema.

La aplicación se encargará de filtrar la información de acuerdo a estos hashtags y así mostrar conversaciones de twitter y además fotografías de Instagram que contengan la información en su metadatos para así mostrarlos.



API de Twitter

<https://dev.twitter.com/>

June 11, 2013 Having trouble with your app? API v1 is retired and no longer functional. | Read more →

Developers API Health Blog Discussions Documentation Search Sign in

More downloads for your app with Twitter Cards

Twitter Cards offer a fast and easy way to grow your user base for mobile apps. Simply add some new markup to your pages: when users tweet links to your domain, Cards will let other users viewing those Tweets to download and launch your app across a number of mobile platforms.

[Learn More](#)

Estos son métodos que nos brindan la facilidad de ejecutar instrucciones de bajo nivel en los servidores de twitter para así realizar búsquedas con algún criterio establecido en todos sus tweets² recientes, esta se encarga de devolver la información relacionada a la búsqueda para poder ser utilizada por la aplicación para desarrollar sus resultados.

Twitter ofrece tres APIs de los cuales se utilizará:

Search API: Permite el acceso al motor de búsqueda de twitter para así buscar publicaciones recientes (últimos 7 días), los cuales pueden ser utilizados a conveniencia del desarrollador.

En la aplicación se utilizará el Search API, con el cual se realizarán las búsquedas en los tweets más recientes (7 días) y así mostrar la cantidad indicada en pantalla con el criterio de búsqueda indicado.

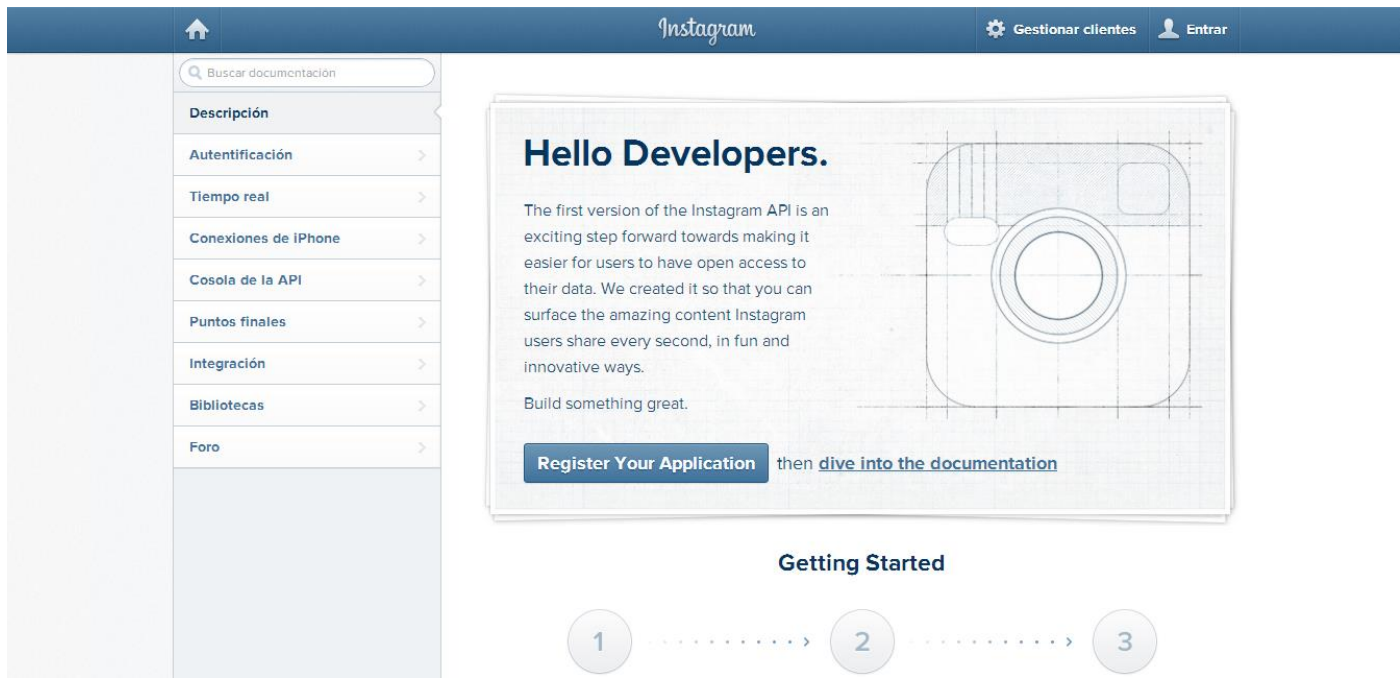
² Publicación o una actualización de tu estado en Twitter. Tweet proviene del sonido que emiten los pájaros, para los que no lo saben el nombre de Twitter significa parlotear (la forma en cómo se comunican los pájaros).



Instagram
API

API de Instagram

<http://instagram.com/developer/>



Al igual que twitter el API de Instagram ofrece una manera ágil y fácil de realizar consultas, estas se hacen siempre y cuando tengamos un credencial de OAuth³, cada API maneja el suyo por individual y esto permite obtener la autorización para realizar las consultas que se desean hacer con los parámetros establecidos por el usuario.

Forma de obtener el contenido de los APIs

Twitter

Lo primero es cargar la librería de Ruby mediante el código `require 'twitter'` el cual nos permite hacer las consultas correspondientes al API, luego se configura la clase creada llamada `class CTwitter` donde realizamos la configuración de la conexión mediante el código

```
Twitter.configure do |config|
  config.consumer_key = "wMT3UONrMDrjddhZDK4qw"
  config.consumer_secret = "tiz9MRNRjg2agJcTqhLFIitGfuAyjbzPuXF2FoBaTI"
  config.oauth_token = "473447215-xv3DRZb8BxtFJ2MeWshP5pay0JlkyZr0s3uMKTjA"
  config.oauth_token_secret = "g01pRSj1EDf1GegMoxQ2Bat47P8mmTyq7kn7L02k0zE"
```

Con la configuración se le indica al programa las el código de usuario respectivo y claves necesarias generadas por el OAuth para autenticarse y poder realizar la búsqueda (las visualizadas en el código son las necesarias para la aplicación), al entrar en la página <https://dev.twitter.com/> estos tokens son generados y brindados para poder hacer uso de estos en las consultas, las cuales se realizan de la siguiente forma:

```
@resultado = Twitter.search(palabra, :count => cantidad, :result_type => "recent", :lang => "es").results.map
do |status|
  "#{status.created_at}--tweetSeparator--#{status.from_user}--tweetSeparator--#{status.full_text}--
  tweetSeparator--#{status.user.profile_image_url(size=:bigger)}"
```

³ OAuth (Open Authorization) es un protocolo abierto, propuesto por Blaine Cook y Chris Messina, que permite autorización segura de un API de modo estándar y simple para aplicaciones de escritorio, móviles, y web.

Instagram

Lo primero es cargar la librería necesaria con el código `require "instagram"`, una vez hecho esto procedemos a la configuración que se realiza de la siguiente manera:

```
Instagram.configure do |config|
  config.client_id = "410015208"
  config.access_token = "410015208.5b9e1e6.809706cd64d54c0799ff38ba225a7d8d"
end
```

Este código nos permite ingresar el user ID y el token generado por el OAuth de Instagram para así poder realizar las consultas de imágenes referente al tag solicitado. Las llamadas a los resultados se realizan de la siguiente forma:

```
@resultado = Instagram.tag_recent_media (palabra,options={:count=>cantidad})
```

Esto nos permite recibir las fotografías recientes con los tags relacionados a la búsqueda, además de especificar una cantidad de imágenes que se desean mostrar.

Forma de almacenar la información

Una vez que se le hace la solicitud de consulta al API respectivo, se debe proceder a almacenar la información en la aplicación para así poder trabajarla y desplegarla en la página web, para esto utilizamos **Nodos**, estos nodos son los encargados de almacenar la información en los objetos respectivos para así poder desplazarse sobre estos en el caso de Twitter el constructor el objeto queda de la siguiente forma:

```
class NodoTwitter
  attr_reader :autorNombre, :autorImagen, :mensaje, :hora
  attr_writer :sigNodo, :antNodo
  def initialize(sig, ant, autorN, autorI, mensaje, hora)
    @autorNombre=autorN
    @autorImagen=autorI
    @mensaje=mensaje
    @hora=hora
    @sigNodo=sig
    @antNodo=ant
  end
end
```

Como podemos ver en la zona marcada con amarillo, lo que se hace es crear nodos que conocen a los anteriores y a los siguientes, de esta forma creamos **Listas doblemente enlazadas** y así podemos recorrer la información para ir organizándola en la página web y presentar los resultados. En el caso de Instagram el constructor queda de la siguiente manera:

```
class NodoInstagram
  attr_reader :autorNombre, :imagen, :hora, :sigNodo, :antNodo
  def initialize(sig, ant, autorN, imagen, hora)
    @autorNombre=autorN
    @imagen=imagen
    @hora=hora
    @sigNodo=sig
    @antNodo=ant
  end
end
```

Lenguajes utilizados

Ruby on Rails



Para la creación de la aplicación se utiliza el lenguaje Ruby on Rails (<http://www.ruby-lang.org/>), con el cual se diseña la estructura principal de la aplicación y es aquí donde se crean las clases que son las encargadas de realizar las consultas y recibir y enviar parámetros a los servidores, además de los métodos que se encargan de todo el manejo de la información, cuales son los datos que serán mostrados, además de métodos que se encargan de separar y seleccionar la información para así enviarla por medio de SINATRA el cual recibe el código HTML y se encarga de mostrarlo en la página creada.

→ SINATRA (Framework)



Sinatra es un framework para aplicaciones web de software libre y código abierto, y Lenguaje específico del dominio escrita en Ruby. Es una alternativa a otros frameworks para aplicaciones webs como Ruby on Rails. Sinatra se enfoca en la "rápida creación de aplicaciones web en Ruby con el mínimo esfuerzo."

Sinatra se encarga de tomar el código HTML que se le envía por medio de Ruby y posteriormente lo recibe para crear la página web para así mostrarla al usuario, es una forma fácil y rápida de crear páginas web desde Ruby.

CSS3 (Cascading Style Sheet)



Las hojas de estilo en cascada hacen referencia a un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML, pero también puede ser aplicado a cualquier tipo de documentos XML, incluyendo SVG y XUL.

Para este trabajo se utiliza para darle un mejor aspecto a la página web que se crea y de esta forma poder darle efectos y una presentación más agradable para los usuarios.

JavaScript



JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador.

Para la tarea lo utilizamos para elaborar la página web donde se encontrará el buscador de tweets y de imágenes, esto nos permite trabajar estilos y aspectos, lo cual nos brinda la opción de desarrollar las características visuales de la aplicación.

→ jQuery (Biblioteca de JavaScript)



jQuery es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones (FLV) y agregar interacción con la técnica AJAX a páginas web.

Los lenguajes de programación se utilizan en conjunto para así crear la aplicación web que unifique todos los procedimientos y métodos que nos permitan alcanzar todos los objetivos solicitados en el alcance del programa, si bien la estructura principal es basada en Ruby on Rails también los demás lenguajes son de suma importancia a la hora de desplegar la información en pantalla para ser interpretada y utilizada por los usuarios, ya que es este tipo de aplicaciones la importancia de la interfaz gráfica de usuario es un factor de peso para el éxito de la misma.

Clases y Métodos utilizados

A continuación se hace referencia a las clases utilizadas en los diferentes archivos programados en Ruby, así como los métodos que cada una de ellas contiene, estos serán brevemente explicados para dar una idea de su funcionamiento:

Archivo	Librerías	Clase	Método	Función asignada
main.rb	'rubygems' 'sinatra' 'slim' "instagram" "twitter" 'date'	Clase Main Principal	analizarTextoTwitter()	Recibe el resultado de la consulta al API de Twitter y crea las clases respectivas para manejar la información
			analizarTextoInstagram()	Recibe el resultado de la consulta al API de Instagram y crea las clases respectivas para manejar la información
CInstagram.rb	"rubygems" "instagram"	CInstagram	Initialize()	Constructor que se encarga de configurar el API además de hacer la solicitud al mismo.
			imprimirResultado()	Imprime el valor de la variable resultado en pantalla.
			obtenerResultado()	Retorna el valor almacenado en la variable resultado.
CTwitter.rb	'rubygems' 'twitter'	CTwitter	Initialize()	Constructor que se encarga de configurar el API además de hacer la solicitud al mismo.
			imprimirResultado()	Imprime el valor de la variable resultado en pantalla.
			obtenerResultado()	Retorna el valor almacenado en la variable resultado.
listaInstagram.rb	No existen librerías necesarias	ListaInstagram	Initialize()	Constructor encargado de generar la lista doblemente enlazada vacía.
			Insertar()	Inserta un nuevo nodo a la lista doblemente enlazada utilizada para almacenar los resultados de la consulta.
			obtenerHtml()	Se encarga de tomar la lista de nodos y crear el código HTML para enviarlo a Sinatra y pueda ser utilizado en la página web.
listaTwitter.rb		ListaTwitter	Initialize()	Constructor encargado de generar la lista doblemente enlazada vacía.
			Insertar()	Inserta un Nodo en la lista para almacenar los resultados.
			obtenerHtml()	Se encarga de tomar la lista de nodos y crear el código HTML para enviarlo a Sinatra y pueda ser utilizado en la página web.
nodoIn.rb		NodoInstagram	Initialize()	Constructor que se encarga de generar el nodo con los datos necesarios.
			obtenerHtml()	Obtiene el código HTML de cada nodo para aplicarlo en la página web.

Archivo	Librerías	Clase	Método	Función asignada
nodoTw.rb		NodoTwitter	Initialize()	Constructor que se encarga de generar el nodo con los datos necesarios.
			obtenerHtml()	Obtiene el código HTML de cada nodo para aplicarlo en la página web.
			sigNodoCambiar()	Método utilizado para avanzar en la lista de nodos.
			obtenerSiguiente()	Retorna el objeto nodo siguiente al actual.
			obtenerAnterior()	Retorna el objeto anterior al actual.
			obtenerAutor()	Retorna el valor del nombre del autor.

Análisis de resultados

Objetivos Alcanzados

Se muestra una lista de objetivos alcanzados los cuales fueron solicitados para la tarea programada:

- ✓ La aplicación deberá tener un formulario de búsqueda, en el cual los usuarios podrán especificar un criterio de búsqueda (los hashtags).
- ✓ La aplicación deberá usar el API de búsqueda de Twitter e Instagram, y obtener el contenido (tweets y fotos, respectivamente) que concuerden con el criterio de búsqueda.
- ✓ El número de tweets y fotos que se obtenga por cada búsqueda podrá ser configurable, ya sea desde la misma aplicación, o desde un archivo de configuración del lado del servidor.
- ✓ Mostrar la siguiente información
 - Tweets: texto, timestamp, usuario, foto usuario
 - Fotos: foto, timestamp, usuario
- ✓ La información de los resultados la deberán manejar a nivel interno mediante objetos.
- ✓ El programa deberá ser una aplicación web, para lo cual se usará un framework de creación de desarrollo de web apps llamado Sinatra (<http://www.sinatrarb.com/>).
- ✓ La aplicación deberá ser desarrollada usando Ruby como lenguaje de programación, y Sinatra como framework para manejo de aplicaciones web. Además deberá poder ejecutarse en el sistema operativo Linux.

Objetivos no Alcanzados

Existe un pequeño inconveniente con la cantidad de imágenes solicitadas al API de Instagram, ya que en varias de las pruebas realizadas notamos que la cantidad de resultados mostrados no concordaba con la cantidad solicitada, muchas de las veces se mostraba una cantidad menor o mayor a la establecida, al parecer esto se debe a un error en el filtro del API que se encarga de definir la cantidad de imágenes que va a consultar, muchas de las veces se debió a que no había la cantidad suficiente de imágenes para complacer la consulta por lo que devuelve únicamente la cantidad encontrada. Se plantea una solución en el código fuente de Ruby pero al ejecutarla sigue sin la capacidad de corregir el error por lo que se indica en esta sección de los objetivos no alcanzados del programa.