

Instituto Tecnológico de Costa Rica

Escuela de Computación

Administración en Tecnologías de Información

Lenguajes De Programación
Prof. Andrei Fuentes

TAREA PROGRAMADA #4

TWEET MY BAND

Andrey Bolaños Sandoval
Yessenia Montiel
Christopher Vasquez

Cartago, 2012

INDICE

1.	Descripción del problema	3
2.	Diseño del programa.....	4
2.1.	Procesamiento de la información	4
2.2.	Aplicación Web	5
2.3.	BandCamp Developer API y el BandCamp API Extendido.....	5
2.4.	Widget Social :: Tweet Button	5
3.	Librerías utilizadas.....	5
4.	Análisis de resultados	6
4.1.	Objetivos básicos	6
4.2.	Objetivos adicionales (puntos adicionales).....	6
4.3.	Áreas de mejora.....	6
5.	Manual de usuario	7
6.	Conclusión personal.....	10
7.	Código Fuente	10
8.	Referencias.....	10

1. Descripción del problema

Se requiere crear un programa en RUBY que obtenga datos de 2 APIs con el fin de alimentar una aplicación que realiza búsquedas de bandas/álbumes/canciones basado en un “tag” de género y país. Además de esto debe permitir “tweetear” los resultados obtenidos siguiendo un formato preestablecido. El grupo determino que el proyecto debía dividirse en 3 módulos y su integración:

- a) Conexión, obtención y representación de los datos solicitados del API de BandCamp.com
- b) Interfaz gráfica
- c) Conexión a Twitter para postear los resultados obtenidos

A continuación se detallan algunas de los problemas o lecciones aprendidas del proyecto:

a) Conexión con BandCamp

- i. Originalmente se pensó en utilizar el Developer API de la página, pero actualmente el Dev Team no esta ofreciendo llaves por lo que no era posible obtener los datos deseados
- ii. Aun pudiendo tener un acceso temporal al API, el servicio no brinda soporte a búsqueda por etiquetas, lo que nos llevo a concluir que se requería leer del HTML directamente de las páginas de etiquetas y procesarlo adecuadamente para obtener los datos deseados
- iii. Se desarrollo un sistema externo que denominamos “BandCamp Extended API” que básicamente añade la funcionalidad de búsquedas por tags
- iv. Se presentaron problema de Cross-Domain, por lo que fue necesario agregar soporte para el formato JSONP en el API extendido

b) Conexión con Twitter

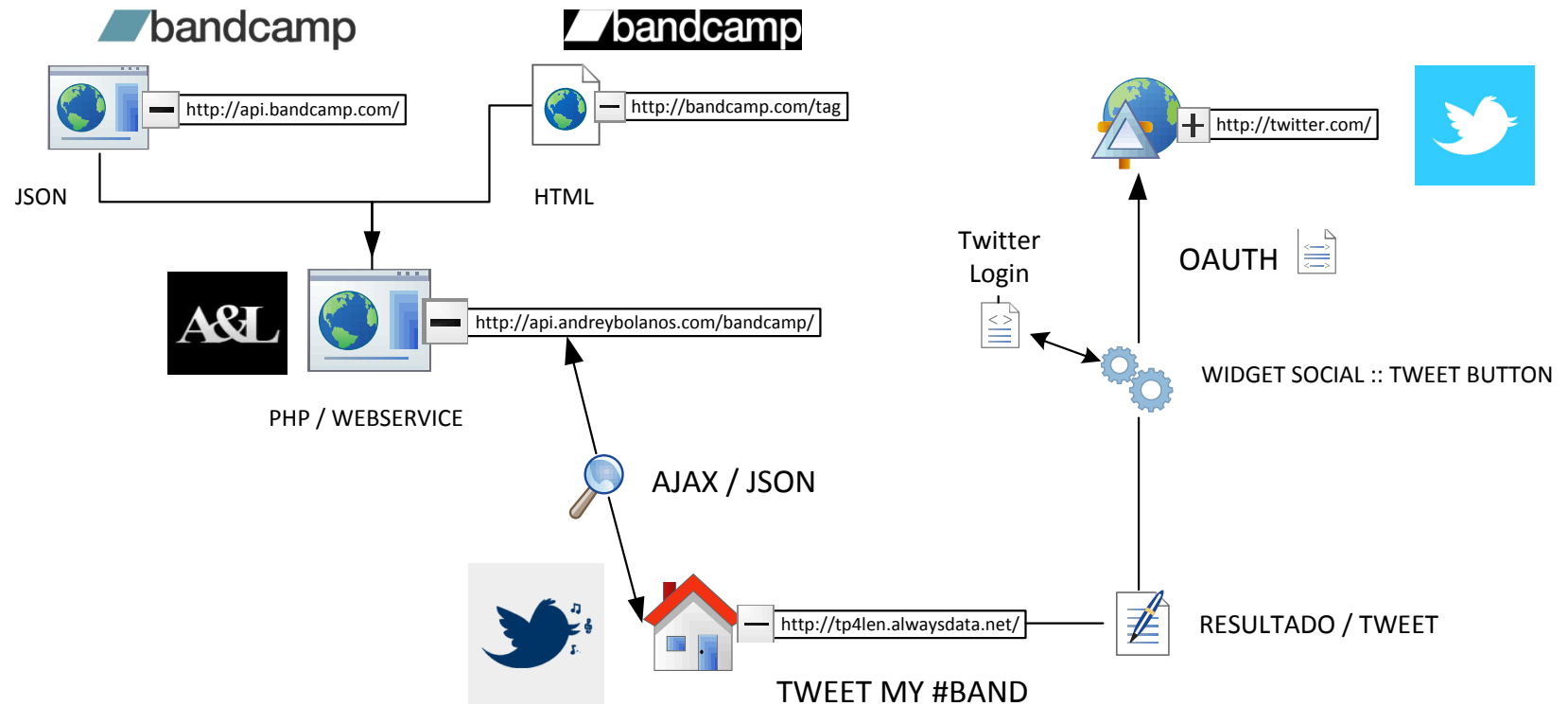
- i. Originalmente se solicitaba utilizar la conexión OAuth de Twitter e incluso se desarrollo un prototipo que aplicaba este método de autenticación satisfactoriamente enviando un tweet.
- ii. En la fase de desarrollo se descubrieron los “Tweet Buttons” que son widgets sociales mucho más fáciles de implementar y que cumplen con la funcionalidad solicitada

c) Interfaz gráfica

- i. Se procedió a evaluar diferentes propuestas gráficas de los miembros y se determinó que la aplicación sería desarrollada para web, ya que esto permitiría evadir el problema de multi-plataforma (Windows o unix) y además le daría mayor visibilidad e usabilidad al sistema.
- ii. Para tal efecto se procedió a utilizar Ruby On Rails (framework) lo cuál facilitó muchísimo la maqueta inicial del sitio, pero presento problemas reiterativos a la hora de configurar todos los ambientes de trabajo (siempre ocurrían errores por gemas no instaladas).
- iii. Como último problema, fui imposible implementar la estructura de clases adecuadas en MVC que permitieran hacer las consultas por AJAX. El equipo aún debe capacitarse más en Ruby y en patrones de diseño antes de proceder a trabajar de lleno con este framework.
- iv. Debido al tiempo restante, se procedió a programar una solución alternativa que utilizaría javascript (jQuery) para unir los módulos restantes.

2. Diseño del programa

2.1. Procesamiento de la información



2.2. Aplicación Web

En la Fase 1 (etapa de planeación) se tomó en consideración todas las características del sistema y se determinó que todas requerían acceso a internet y que por ende su uso sería más adecuado desde una plataforma web. Esto además conlleva a una serie de ventajas:

- Soporte multi-plataforma (es indiferente del sistema Windows o Unix). Se chequeo que el sistema funcionara correctamente en los buscadores web más usados (Chrome, Firefox, IE)
- Uso intensivo de API's procesados mediante Javascript en el frontend
- Implementación de widgets que simplifican el acceso a la cuenta de Twitter

2.3. BandCamp Developer API y el BandCamp API Extendido

Según los problemas descritos en el apartado "Descripción del problema", se desarrollo un servicio web externo que recorre el HTML de las paginas de tags y obtiene más información de cada resultado consultando el API de BandCamp.

Esto es de gran utilidad ya que el sistema normaliza las consultas utilizando el formato JSON y además aumenta la flexibilidad de desarrollo si se agregan más módulos y funciones al API extendido.

Para mayor información visitar: api.andreybolanos.com

2.4. Widget Social :: Tweet Button

A pesar de que se logro enviar un tweet desde el backend en ruby, se considero que no era necesario agregar la lógica desde el backend, ya que este widget desarrollado por el Twitter Dev Team ya provee las funcionalidades solicitadas de "Login por OAuth" y "Tweet".

Otro aspecto a considerar es que el soporte del sistema a largo plazo, ya que actualmente se usa OAUTH pero un futuro esto podría cambiar (como ocurrió con el anterior sistema por CURL). Nuestra solución brinda mayor soporte ya que es un widget desarrollado por Twitter.

3. Librerías utilizadas

#require	Descripción
RUBY	
gem :: json	Core de Ruby encargado de parsear formatos JSON
gem :: net/http	Core de Ruby utilizado para capturar contenido de respuestas HTTP
gem :: oauth	Core de Ruby utilizado para realizar autenticaciones usando el protocolo OAUTH
gem :: twitter	Librería de Twitter, representa un cliente de acceso a la aplicación
WEB	
jQuery 1.7.2	Librería Javascript utilizada para la manipulación del DOM

4. Análisis de resultados

4.1. Objetivos básicos

Descripción del objetivo	Resultado
FASE 1	
Conectarse al API de BandCamp desde Ruby	OK
Conectarse al API extendido de BandCamp desde Ruby	OK
Propuesta de la interfaz gráfica	OK
Enviar un tweet desde Ruby	OK
FASE 2	
Implementar la interfaz gráfica en Ruby on Rails	OK
Cargar los resultados del API extendido de BandCamp en un modelo de Ruby on Rails	OK
Implementar la funcionalidad de búsqueda (botón search) aplicando el patrón MVC de Ruby on Rails	<i>No se entendió como implementar vista parciales para refrescar los resultados</i>
Implementar la funcionalidad de tweet (link de tweet) aplicando el patrón MVC de Ruby on Rails	<i>Los datos se almacenaban en sesión, pero no se logro iterar y tweetear cada resultado</i>
FASE 3 <i>No fue posible usar el patrón MVC, se preparo una solución alterna de la interfaz</i>	
Crear la interfaz gráfica desde Ruby	OK
Administrar el contenido dinámico con jQuery (consulta o búsqueda, e inserción de los resultados en el DOM)	OK
Implementación del widget Tweet Button	OK

4.2. Objetivos adicionales (puntos adicionales)

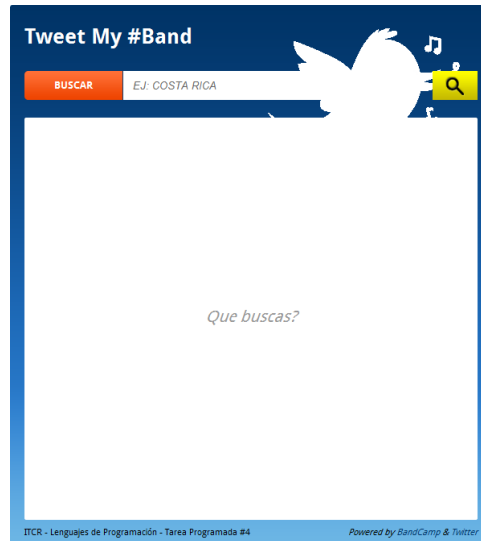
Descripción del objetivo	Resultado
Interfaz gráfica	SI

4.3. Áreas de mejora

Descripción del área de mejora	Resultado
Patrón MVC	Se intento utilizar el potencial del framework Ruby on Rails, pero fue muy difícil lograrlo por ser la primera vez que se utilizaba este lenguaje.
Rendimiento del API y consultas	Se considera que algunas consultas poseen un nivel alto de latencia por lo que sería importante manejar algún nivel de cache en el servidor o implementar algún parámetro para reducir los tiempos.

5. Manual de usuario

- a) Acceder al sitio web <http://tp4len.alwaysdata.net/>



- b) Se muestra un ejemplo "Ej: Costa Rica". Escribir ahí el **tag** a buscar (país o genero)



- c) Se carga una lista con los resultados obtenidos



d) Para obtener más información del resultado, click en el link **+ información**

+ información del **album**

The screenshot shows the album page for 'Grape Soda EP' by Grape Soda! on the Triple J Unearthed website. At the top, a banner reads 'we are NOW called...TUNDRA' with the Facebook link 'WWW.FACEBOOK.COM/TUNDRABAND'. The album title 'Grape Soda EP' is displayed, along with a play button and a progress bar for the track 'Nauseous'. Below this, it says 'Digital Album' and offers an immediate download in MP3, 320, FLAC, or other formats. The price is listed as 'Buy Now \$0.50 AUD or more'. A 'Share / Embed' section lists the tracks: 1. Nauseous (05:05), 2. James and the Giant Peach (04:30), 3. Omnibus to the moon (04:09), and 4. Mind (03:18). It also mentions the release date (25 December 2011) and the genre (pop Brisbane). On the right, there is a description of the band, social media links for Facebook and Twitter, and a 'contact / help' section.

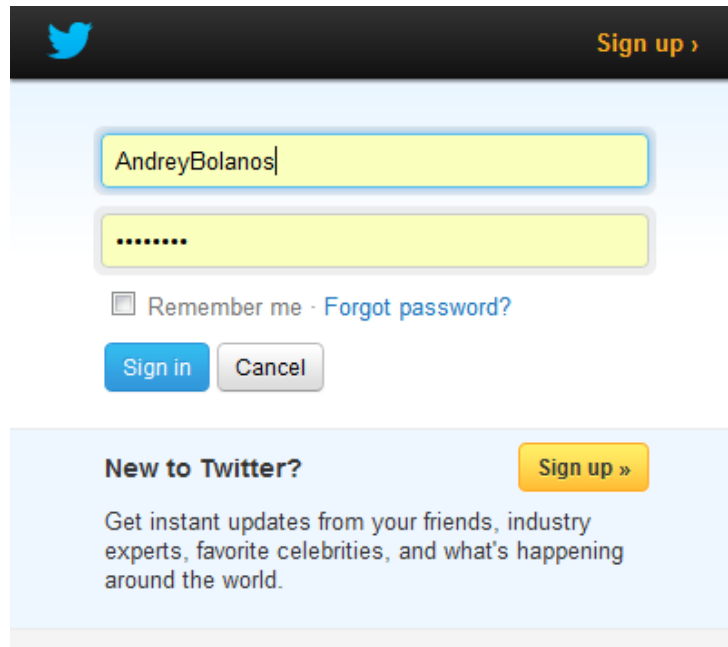
e) Para descargar click en **Bajar Gratis** o puede comprarla haciendo click en **Comprar**

This block contains two buttons for purchasing or downloading the album. The top button is labeled 'COMPRAR' and features a musical note icon. The bottom button is labeled 'BAJAR GRATIS' and also features a musical note icon.

f) Para *tweetear* un resultado, click en el botón **Tweet**

i. Si no se encuentra loggeado, Twitter le solicitará acceder a su cuenta

A Twitter 'Tweet' button with a blue bird icon and a counter showing '290' tweets.



The image shows the Twitter login and sign-up interface. At the top, there is a black header with the Twitter logo on the left and a "Sign up >" link on the right. Below the header, there are two yellow input fields. The first field contains the text "AndreyBolanos". The second field contains a series of dots, representing a password. Below the password field, there is a checkbox labeled "Remember me" and a link "Forgot password?". Below these, there are two buttons: "Sign in" (blue) and "Cancel" (gray). At the bottom, there is a section titled "New to Twitter?" with a yellow "Sign up »" button. Below the title, there is a paragraph of text: "Get instant updates from your friends, industry experts, favorite celebrities, and what's happening around the world."

- ii. Revisar el mensaje a twittear, click en Tweet



The image shows the Twitter tweet composition interface. At the top, there is a black header with the Twitter logo on the left and a profile picture and name "ASL AndreyBolanos" on the right. Below the header, there is a section titled "Share a link with your followers". Below the title, there is a text area containing the text: "Yo elegi Grape Soda EP de Grape Soda! que es PAID http://grapesodaus.bandcamp.com/album/grape-soda-ep via @TweetMyBand". Below the text area, there is a "53" and a "Tweet" button.

6. Conclusión personal

A pesar de las dificultades que se nos presentaron con la implementación del sistema en Ruby on Rails, la tarea fue de gran provecho ya que aprendimos sobre:

- Acceso a API's y procesamiento de datos en formato JSON
- Métodos de autenticación (en este caso el protocolo OAuth)
- Implementación de widgets sociales

Como grupo tenemos un reto muy importante que es comprender el patrón MVC / REST y aprender a utilizar todo su potencial mediante el framework Ruby on Rails.

7. Código Fuente

GITHUB	https://github.com/Yesenia1/TEC-2012-Lenguajes-P4
--------	---

8. Referencias

- <http://bandcamp.com/developer>
- <http://api.andreybolanos.com/>
- http://es.wikipedia.org/wiki/Servicio_web
- <http://www.json.org/>
- <http://es.wikipedia.org/wiki/JSON>
- <http://api.jquery.com/jquery.ajax/>
- http://en.wikipedia.org/wiki/Cross-domain_solution
- <http://developer.yahoo.com/ruby/ruby-json.html>
- http://guides.rubyonrails.org/getting_started.html
- <http://rubyonrails.org/documentation>
- <http://api.rubyonrails.org/>
- <http://flori.github.com/json/doc/index.html>
- <http://twitter.rubyforge.org/>
- <http://www.noupe.com/ajax/create-a-simple-twitter-app.html>
- <https://dev.twitter.com/docs/tweet-button>
- <https://dev.twitter.com/docs/auth/oauth/faq>
- <https://dev.twitter.com/docs/auth/oauth>