

Documentación Externa

TP4

TI-3404 Lenguajes de Programación.

## *Búsqueda de Bandas en Bandcamp.*

*Utilizando Ruby, Sinatra.*



Andrés Maroto	201128337
David Murillo	201108648
Adrián Siles	201136841
Ariel Arias	201136814

Profesor: Andrei Fuentes L.

21 Noviembre 2013.

# Tabla de Contenidos

• Descripción del problema.....	3
• Diseño del programa .....	4
• Librerías usadas.....	7
• Análisis de resultados.....	8
• Manual de usuario.....	9
• Conclusión personal.....	20
• Bibliografía Consultada.....	21

# Descripción del Problema

El objetivo de éste proyecto es aplicar los conocimientos aprendidos en el paradigma orientado a objetos realizando una aplicación web que efectúe búsquedas de artistas o grupos musicales que se encuentran en las bases de datos de la página web [www.bandcamp.com](http://www.bandcamp.com). Todo esto utilizando un framework de creación de desarrollo de web apps llamado Sinatra.

La información que se desea adquirir:

Nombre del Grupo

Nombre del Álbum

Si es de paga o no

Url del grupo en la página bandcamp.

Para realizar la búsqueda, el usuario podrá ingresar como parámetro, ya sea el género musical al que pertenece o la ciudad. Mostrando como resultado todos los primeros diez grupos que pertenezcan a los tags mencionados antes.

Cabe destacar que no se podrá contar con el API de bandcamp por lo que se deberá de implementar un algoritmo que recorra el código fuente de la página y extraiga la información requerida mediante el manejo de la orientación a objetos.

Cada uno de los diez resultados deberá tener la capacidad de conectarse con twitter, pero tomando en cuenta la restricción de 140 caracteres, por lo que de ser necesario se deberá de truncar los datos.

# Diseño del Programa

El programa se compone de un archivo .rb, que va a estar escrito en ruby y va a contener todas las funcionalidades y en el cual por medio de sinatra se puede interactuar con páginas html, cuya extensión va a ser un .erb, ya que va a permitir el uso de código de ruby en estos archivos. La única diferencia es el uso de `<% %>` como etiquetas para utilizarlo.

Para realizar el algoritmo de búsqueda definimos primero la clase con sus respectivas variables, esto para que éstas almacenen el nombre del grupo musical, una imagen del mismo, el nombre del álbum, si éste está a la venta o es gratis y el url de bandcamp relacionado con la banda.

Para la captación del tag de búsqueda al oprimir el botón respectivo llama a un método dentro del programa principal llamado “get”, que convierte lo que viene de la casilla de texto y lo convierte en un string, para posteriormente guardarlo en una variable y enviarlo a la función de búsqueda.

Para la búsqueda por un tag específico, se recibe y se encadena con un string para que este sea el url donde se irá a buscar el html necesario para extraer los resultados. Para utilizar este string como una dirección de url y extraer su html se utiliza `Hpricot(open(link))`, esto para convertir el resultado de procesar el url y obtener el html en un array por el cual se operará en busca de ciertas etiquetas para obtener la información de cada resultado de la búsqueda que serán almacenados en una lista de objetos de la clase resultado, los cuales contienen toda la información respectiva a cada resultado.

Al haber terminado la función de búsqueda, si se ha efectuado de manera correcta, obteniendo algún resultado, se obtendrá una lista de objetos, por

lo que pasará a la función que coloca los primeros diez resultados con su respectiva información(atributos). En caso de haber menos de diez, la función reemplaza la información de los contenedores vacíos por “ ”. De ésta manera al abrir la página de resultados sólo la información que se desea.

En caso de que el método de búsqueda no despliega ningún resultado, la página principal se desplazará hacia la página de error donde se observa un mensaje que dice que no hay resultados y un botón que regresa hacia la principal nuevamente.

En cada resultado estará la opción de escribir en un tweet que contendrá la información del url de banda y el usuario decidirá si publicarlo o no. Esto se logra mediante la implementación de un widget de twitter preconstruído.

# Librerías Usadas

Se necesitó de la instalación de una serie de “gems” necesario para la elaboración correcta del proyecto como los que vemos a continuación:

- Sinatra
- Rubygems
- erb
- hpricot

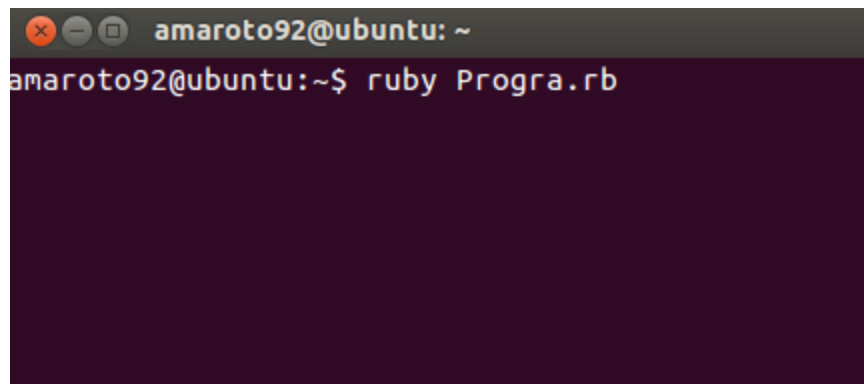
## Análisis de Resultados

Se logra desarrollar el proyecto por completo, con funcionalidad a un 100% según lo solicitado en la tarea.

# Manual de Usuario

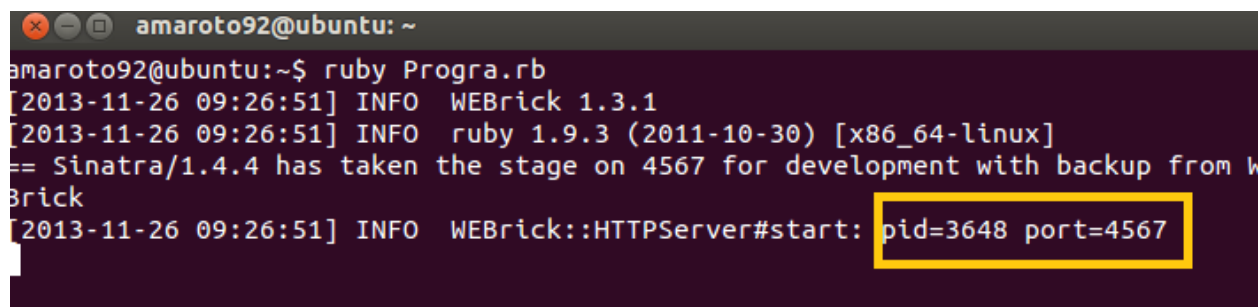
Lo primero que tenemos que hacer es ponernos en el directorio donde se encuentra el archivo Progra.rb y asegurarse que la carpeta “Views” esté en el mismo directorio.

Luego abrimos la terminal y abrimos el programa de la siguiente manera:

A terminal window with a dark background. The title bar shows 'amaroto92@ubuntu: ~'. The prompt is 'amaroto92@ubuntu:~\$' and the command 'ruby Progra.rb' has been entered.

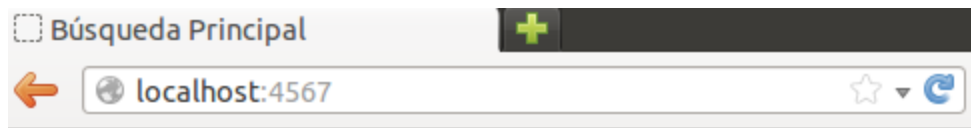
```
amaroto92@ubuntu: ~  
amaroto92@ubuntu:~$ ruby Progra.rb
```

Cuando le damos enter, el programa nos dirá en qué puerto del local host estará montado el programa y su id de proceso.

A terminal window showing the output of the 'ruby Progra.rb' command. The output includes version information for WEBrick and Ruby, and a message from Sinatra. The last line, '[2013-11-26 09:26:51] INFO WEBrick::HTTPServer#start: pid=3648 port=4567', is highlighted with a yellow box.

```
amaroto92@ubuntu:~$ ruby Progra.rb  
[2013-11-26 09:26:51] INFO WEBrick 1.3.1  
[2013-11-26 09:26:51] INFO ruby 1.9.3 (2011-10-30) [x86_64-linux]  
== Sinatra/1.4.4 has taken the stage on 4567 for development with backup from WEBrick  
[2013-11-26 09:26:51] INFO WEBrick::HTTPServer#start: pid=3648 port=4567
```

Ahora que ya está corriendo el programa procedemos a abrir cualquier navegador y en la barra de dirección ponemos lo siguiente:



Esto nos desplegará la página principal del programa. En la esquina inferior derecha estará el campo de texto con su respectivo botón para que pueda insertar su parámetro de búsqueda.

## Buscador de grupos de Bandcamp

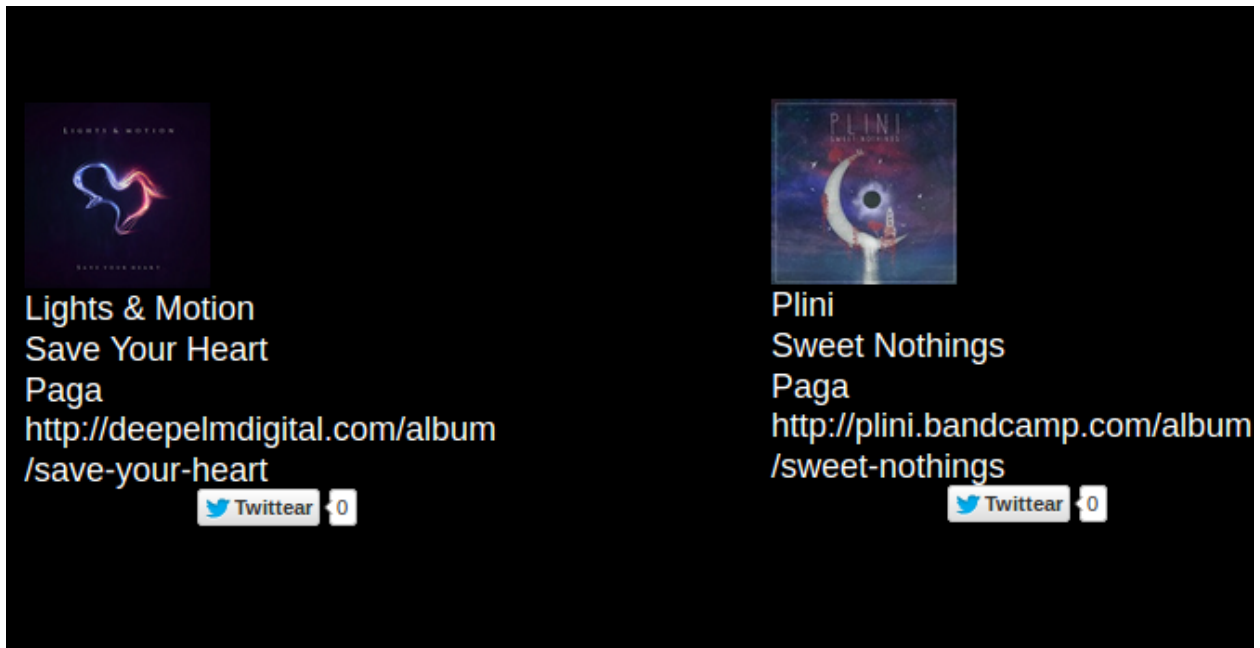
Este es un motor de búsqueda de los grupos musicales que en la pagina de bandcamp, que se implementa ingresando e ya sea el género musical al que pertenece o la ciudad.

Escriba el genero o la ciudad

Buscar

Luego escribimos cualquier búsqueda, por ejemplo “indie rock” y le damos buscar. Esto nos abrirá una nueva página donde despliega los resultados encontrados y la información requerida.





Si deseamos twitrear el resultado pulsamos el botón de Twitter y se nos aparecerá el cuadro para poder loggearse a Twitter y twitrear el resultado.

The image is a screenshot of the Twitter login page. The browser address bar shows 'https://twitter.com/intent/tweet?original\_referer=http%'. The Twitter logo is in the top left, and 'Sign up >' is in the top right. The main heading is 'What's happening?'. Below it is a text input field containing 'Lights & Motion Save Your Heart Paga'. Below the text field are two input fields: 'Username or email' and 'Password'. To the right of the 'Username or email' field is the number '104'. To the right of the 'Password' field is a blue button labeled 'Sign in and Tweet'. At the bottom left, there is a checkbox labeled 'Remember me' and a link labeled 'Forgot password?'.

Es importante recalcar que cuando ya queremos cerrar el programa tenemos que ir nuevamente a la terminal y teclear la combinación de teclas para cerrar el proceso y liberar el puerto. Normalmente es control-c.

```
^C== Sinatra has ended his set (crowd applauds)
[2013-11-26 09:38:40] INFO going to shutdown ...
[2013-11-26 09:38:40] INFO WEBrick::HTTPServer#start done.
```

## Conclusión Personal

Para el desarrollo de esta cuarta tarea programada, se debía trabajar con el lenguaje de programación ruby y además de utilizar sinatra como Por otro lado el uso de la herramienta Github fue fundamental en el desarrollo de esta tarea ya que facilita el proceso a la hora de programar grupalmente. Y brinda opciones para realizar el “branching” de diferentes versiones del proyecto. Gracias a este control de versiones es que facilita la programación en un grupo porque al almacenar todas las versiones del código que se han incluido en el repositorio, resulta sencillo poder observar y comprender los cambios que realiza algún compañero y de ser necesario se pueden encontrar las versiones anteriores a un cambio. Esto sumándole que nuestro grupo de trabajo era de 4 personas y nos ayudó a organizarnos de la mejor manera.

Para desarrollar el programa debíamos investigar acerca del lenguaje de ruby y como se podría analizar un html, además se debía investigar cómo utilizar sinatra para dar los resultados que el programa debía retornar. Gracias al desarrollo de esta tarea logramos ampliar y mejorar nuestros destrezas en el lenguaje de Ruby, Html y Sinatra.

Es muy importante destacar que también se pudo analizar y comprender las formas de resolver el problema que propone cada compañero, y con esto poder dar con la mejor combinación de ideas y obtener el mejor resultado.

## Bibliografía Consultada

- IM Creator - Start (s. f.). Recuperado el 21 de Noviembre del 2013, de <http://app.imcreator.com>
- MySpace Codes (s. f.). Recuperado el 21 de Noviembre del 2013, de [http://www.quackit.com/myspace/myspace\\_code\\_generators/](http://www.quackit.com/myspace/myspace_code_generators/)
- Ruby Tutorial - Learn Ruby (s. f.). Recuperado el 21 de Noviembre del 2013, de <http://rubylearning.com/>
- Sinatra: Documentation (s. f.). Recuperado el 21 de Noviembre del 2013, de <http://www.sinatrarb.com/documentation.html>
- Ruby Documentation. Recuperado el 21 de Noviembre del 2013, de <http://ruby-doc.org/core-2.0.0/String.html>