# Macintosh HD:Users:gabrielfernandez:Downloads:tec_logo.png

Tarea Programada IV

**Lenguajes de programación | II semestre**

Integrantes

ANdrei AMador SAlazar

Cinthya Contreras Vargas

Gabriel Fernández Vargas

Roger Vega Rojas

2013

Tabla de contenido

[0](#_Toc372832665)

[Descripción del problema. 2](#_Toc372832666)

[Diseño del problema 3](#_Toc372832667)

[Librerías Usadas 4](#_Toc372832668)

[Manual de Usuario Chiqui Chiqui 5](#_Toc372832669)

[Requerimientos: 6](#_Toc372832670)

[Análisis de Resultado: 14](#_Toc372832671)

[Conclusiones 15](#_Toc372832672)

[Bibliografía 16](#_Toc372832673)

# Descripción del problema.

El sistema consiste en una aplicación web que será desarrollada en el lenguaje de programación Ruby junto con html por lo que se utilizará como herramienta auxiliar **Sinatra** como framework, la aplicación permitirá buscar música ya sea por género musical o por zona, los datos serán tomados del sitio web **Bandcamp (**[**www.bandcamp.com**](http://www.bandcamp.com)**)**, el cual cuenta con la limitación de que el API de la página no se encuentra disponible; por lo que se deberá desarrollar un algoritmo para obtener el código de fuente resultado de la búsqueda y extraer de forma manual los dados estos son:

* Nombre del grupo.
* Nombre del álbum.
* Si se debe de pagar o es gratuito.
* Url de Bandcamp del grupo.

La cantidad de resultados retornados de la búsqueda se almacenará en un archivo de configuración con el fin de que sea modificable la cantidad.

Además por cada resultado obtenido de la búsqueda se deberá ofrecer al usuario la opción de publicar un tweet por lo que se debe de realizar el proceso de autorización de Twitter, el tweet debe de contener toda la información que retorne la función de búsqueda en BandCamp, se deberá de respetar la restricción de 140 caracteres por tweet; de lo contrario se truncarán los datos.

# Diseño del problema

Utilización de html: Para el diseño de la sencilla página web utilizamos los comandos más básicos del lenguaje html, lo que nos permitió utilizar con certeza lo que específicamente necesitábamos para la implementación de la solución computacional.

• Nombre y diseño de la página: Quisimos utilizar esta tarea para rescatar un poco las características del costarricense mediante el lenguaje típico utilizado en nuestro país, por esta razón el título de la página tiene ese lema respaldado por el logo principal.

• Utilización de librerías complementarias: Su necesario recurrir al uso de librerías enlazadas con Ruby para poder desarrollar la aplicación, ya que ciertos procedimientos Ruby no puede realizarlos de manera exclusiva.

• Utilización de Sinatra: Siguiendo la recomendación del profesor utilizamos Sinatra y todo su contenido funcional y documentado para poder conocer su uso e implementación en el diseño de páginas web mediante Ruby y Html.

• Creación de funciones: los procesos de búsqueda en la página de Bandcamp se realizan mediante la ejecución fe una serie de funciones enfocadas cada una en realizar un proceso de búsqueda dentro del sitio web. Descripción breve del algoritmo El algoritmo diseñado para realizar esta aplicación consta de las siguientes etapas:

• Ingreso del dato a buscar: en este proceso se debe recolectar el dato ingresado por la persona en la caja de búsqueda de la página localhostxxxx.

• Envió del dato a buscar: Al presionar **enter** o dar **click** en “Buscar” se envía ese dato a la función encargada de iniciar el proceso de búsqueda.

• Búsqueda de información: Con el dato enviado como parámetro a la función “coco3” se da inicio a la búsqueda de datos similares y compatibles con lo requerido.

• Recolección de información: En el momento de recolección de datos, diferentes funciones se encargan de seleccionarlos y clasificarlos por el tipo, permitiendo obtener un resultado de búsqueda ordenado y libre de anomalías o ambigüedades de información de los 10 resultados a presentar.

• Almacenamiento de información: Una vez que se ha creado el arreglo que contiene todos los resultados de búsqueda, se empieza a recorrer dicho arreglo para extraer los datos, reclasificarlos, titularlos y enviarlos a imprimir en la tabla HTML diseñada previamente para dicho proceso.

• Presentación de información: Por medio de una tabla de datos se presentan los 10 resultados obtenidos de la búsqueda enumerados del 1 al 10 con su respectiva cabecera de tabla, imágenes, link de dirección, nombre, autor, álbum etc. Para el diseño del sitio utilizado para realizar las consultas utilizamos dos “frame”, el principal sirve para dar la bienvenida al usuario, ingresar el dato a buscar y esperar mientras se realiza la búsqueda, el segundo presenta los resultados del proceso de búsqueda con la totalidad de la información así como la posibilidad de dar un tweet a algún resultado específico. Se pueden realizar cuantas búsquedas sean necesarias.

# Librerías Usadas

* Sinatra: Sinatra es un lenguaje específico del dominio que permite la construcción de páginas web en Ruby, la aplicación tiene la ventaja de que para poder trabajar en ella no es necesario tanto protocolo; por lo que su uso es bastante sencillo y efectivo.
* RubyGems: Su principal función es gestionar los paquetes de Ruby ya que provee un formato estándar y autocontenido que permite distribuir programas o librerías en Ruby.
* Open-Uri: Es una biblioteca de Ruby que permite abrir un fichero vía http y manipularlo como un fichero normal.
* Hpricot: Es una parser de HTML para Ruby, este está programado en C por lo que su característica principal es la rapidez, hpricot usa expresiones XML PATH Languaje esto permite que la interpretación de los archivos XML sea expresivos y breves.
* Slim: Es un motor de plantillas rápida y ligera. Slim utiliza Temple para el análisis/compilación además está integrado con Tilt por se puede utilizar junto con Sinatra.
* Nokogiri: Es un analizador HTML, XML,SAX y Reader, tiene la capacidad de buscar los documentos a través de XPath o selectores CSS3.
* Twitter Oauth: Es el protocolo de autenticación que permite a los usuarios aprobar la solicitud para actuar en su nombre sin compartir su contraseña.

## Manual de Usuario Chiqui Chiqui

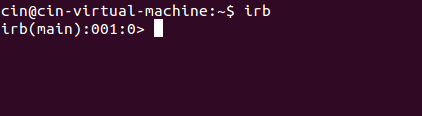


A continuación se mostrará paso a paso como usar “Chiqui Chiqui”.

### Requerimientos:

1. El sistema está desarrollado bajo el sistema operativo Linux por lo que usted debe de poseer alguna extensión de la misma como por ejemplo: Ubuntu, OpenSuse, Fedora, Debian, entre otros.
2. La computadora debe contener el intérprete de Ruby, para instalarlo desde terminal se debe digitar: **sudo apt-get install ruby irb rdoc**

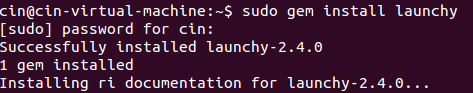
Para comprobar que contamos con el intérprete de Ruby digitamos en terminal: **irb.**



1. Para esta aplicación es necesario el uso de bibliotecas extras. A continuación se mostrarán cómo instalarlos desde consola.

|  |  |
| --- | --- |
| Biblioteca | Comando en Terminal |
| Twitter | sudo gem install twitter |
| Launchy | sudo gem install launchy |
| RubyGems | sugo gem install rubygems |
| Oauth | sudo gem install oauth |
| Nokogiri | sudo gem install nokogiri |
| Sinatra | gem install sinatra |
| Slim | gem install slim |
| Hpricot | gem install hpricot |

Por ejemplo:



Imagén #2: Instalación de Lauchy

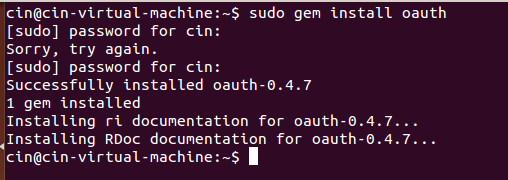


Imagen #3: Instalación de Oauth

1. Correr el programa. Para poder posicionarnos en la carpeta que contiene el archivo utilizamos el comando “cd” y para ver lo que tiene cada carpeta se utiliza “ls”; ya en la carpeta que contiene el programa digitamos:**ruby programa.rb**

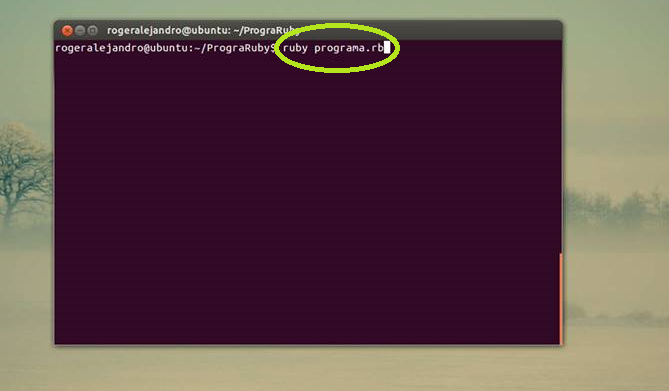
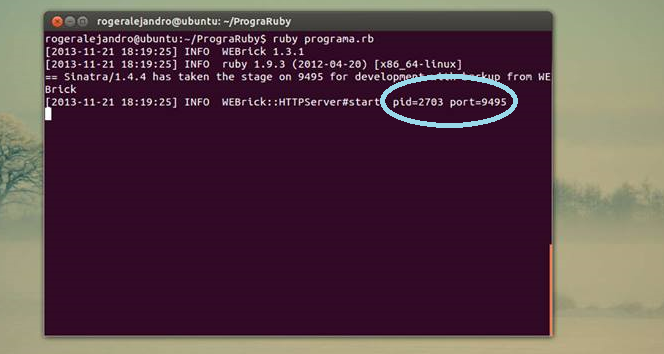


Imagen 1: Ejecución del programa.rb

Cuando se ejecute se mostrará en consola cierta información que será vital para poder mostrar en la red la aplicación “Chiqui Chiqui” como lo es el puerto.



1. Seguidamente nos dirigimos a nuestro buscador y digitamos con el puerto resultado de la ejecución del programa **localhost: 9495**

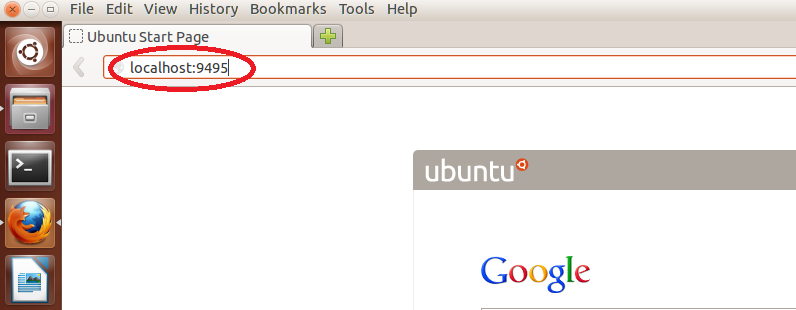


Imagen 2: Nos dirigimos a localhost: 9495



Imagen 3: Página principal “Chiqui Chiqui”

Como se muestra en la imagen anterior la página principal cuenta con la barra buscadora en ella se digitará ya sea la zona o un género musical y se procede hacer click en el botón “Buscar”, inmediatamente se muestran los resultados de la búsqueda.

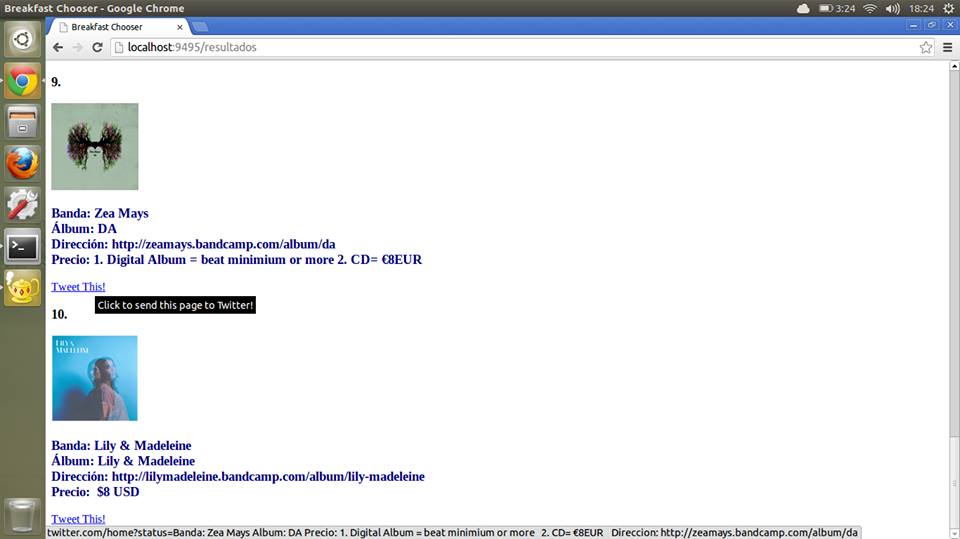


Imagen 4: Ejemplo de búsqueda del género musical

1. Realizar un tweet de algún resultado: Cada resultado de forma individual posee la opción de “Tweet This!” lo que permitirá realizar un tweet con toda la información de un resultado.

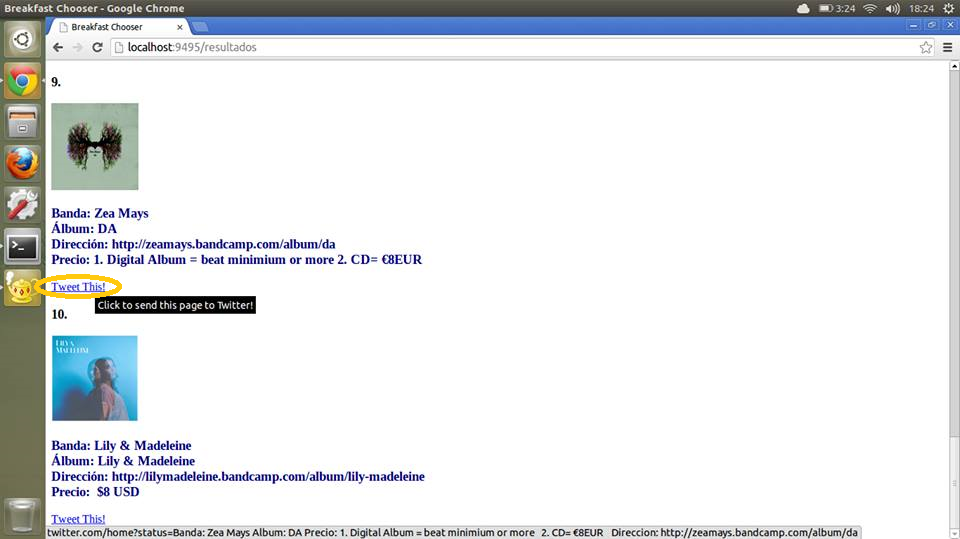


Imagen 5: Opción para realizar un tweet.

1. Login en Twiteer: Si la persona desea realizar un tweet este automáticamente lo direccionará a la página de Twitter. Está ya contará con la información del resultado que usted desee compartir.

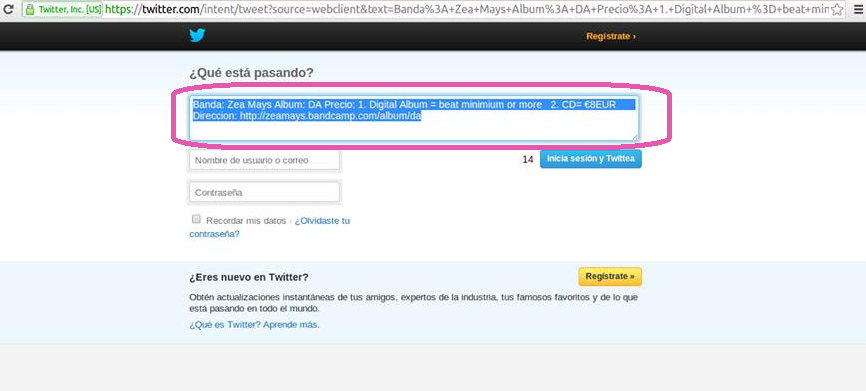


Imagen 6: Página oficial de Twitter para realizar un tweet.

1. El usuario procede a digitar el usuario y la contraseña, además si la persona no posee una cuenta tiene la opción de crearla.



Imagen 7: Login a la cuenta Twitter



Imagen 8: Ejemplo de un tweet de Bandcamp

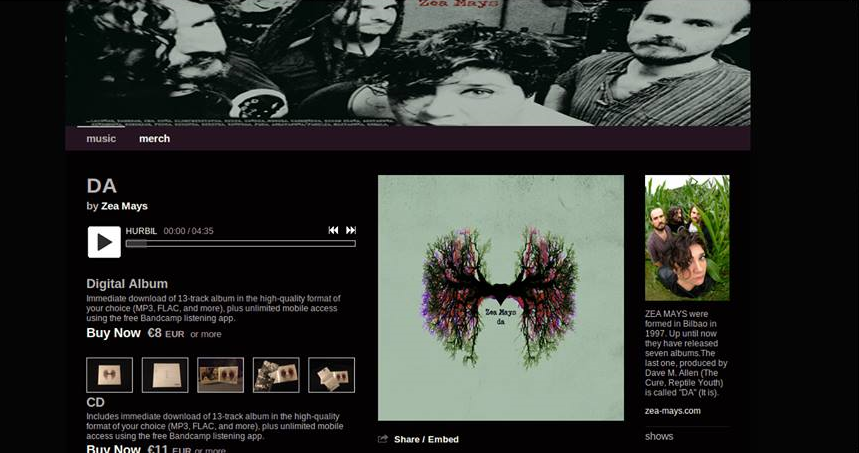


Imagen 9: ejemplo de cuando presione el enlace de twitter como es redirigido a bandcamp

# Análisis de Resultado:

Por medio de una profunda investigación sobre Ruby, Sinatra, HTML, Twitter, entre otros se logró de manera exitosa el desarrollo de la aplicación.

* Se consiguió primeramente la conexión con la página Bandcamp con el propósito de realizar la búsqueda y obtener el código fuente. Se desarrolla el algoritmo que recorre el código hasta extraer todos los resultados de la búsqueda.
* Se alcanzó de manera efectiva la conexión al API de Twitter.
* Se logró una conexión limpia y efectiva de Ruby junto con HTMl gracias a Sinatra y a la librería Slim.
* Se logra desarrollar la aplicación web por medio del framework de Sinatra.
* Es posible mostrarle al usuario todos los datos resultados de la búsqueda estos contempla: nombre del grupo, nombre del álbum, si es gratuita o se debe de pagar algún valor y el URL de Bandcamp.
* Fue posible que por cada resultado dado brindar la opción de realizar un tweet con la información de la búsqueda.

# Conclusiones

Por medio del desarrollo de la tarea programada IV, nuestra curva de aprendizaje con forme se desarrollaba iba en aumento esto debido a que nos vimos obligados a investigar cosas nuevas, nos permitió interactuar con otro lenguaje orientado a objetos diferente a los que estamos acostumbrados como lo son Java y Python aumentando nuestro perfil académico.

Ruby por la simplicidad de su sintaxis, por facilidades que brinda, por su orientación de objetos pura, entre otras características nos permitió una manipulación de datos sencilla y eficiente.

Adquirimos conocimiento sobre sintaxis de html, además aprendimos como utilizar la herramienta Sinatra y conocer las ventajas que esta nos ofrece para poder desarrollar aplicaciones web en Ruby.

También nos permitió entrar en contacto con un medio de comunicación y opinión actual como lo es Twitter, como consecuencia adquirimos el conocimiento de cómo interactuar con el API del sitio y de esa forma poder hacer los tweets que desee el usuario.

Además de forma indirecta adquirimos conocimiento en el área de diseño de aplicaciones web que se la aplicación se trabajó con el fin de que esta sea agradable y de fácil uso para el usuario.

# Bibliografía

* Bibliotecas de Ruby. Consultado en <https://www.ruby-lang.org/es/libraries/>
* HTML Tutorial- (HTML5 ). Consultado de <http://www.w3schools.com/html/default.asp>
* HTML <button> formation Attribute. Consultado en <http://www.w3schools.com/tags/att_button_formaction.asp>
* IMAGE LINKS. Consultado en <http://www.echoecho.com/htmllinks06.htm>
* Nokogiri. Consultado en <http://nokogiri.org/>
* Online HTML Editor. Consultado en <http://www.quackit.com/html/online-html-editor/>
* Sinatra: 29 Links and Resources For A Quicker, Easier Way to Build Webapps. Consultado de <http://www.rubyinside.com/sinatra-29-links-and-resources-for-a-quicker-easier-way-to-build-webapps-1371.html>
* Sinatra, un framework web para Ruby. Consultado en <http://robertoluis.wordpress.com/2012/02/27/sinatra-un-framework-web-para-ruby/>
* OpenURI. Consultado en <http://ruby-doc.org/stdlib-2.0.0/libdoc/open-uri/rdoc/OpenURI.html>
* OAuth FAQ. Consultado en <https://dev.twitter.com/docs/auth/oauth/faq>
* Sinatra. Consultado en <http://www.sinatrarb.com/>
* Singing with Sinatra. Consultado en <http://net.tutsplus.com/tutorials/ruby/singing-with-sinatra/>
* Slim. Consultado en <http://slim-lang.com/>
* Slim 2.0.2.Consultado en <http://rubygems.org/gems/slim>