Documentación Escrita
Cuarto Proyecto Programado

Instituto Tecnológico de Costa Rica
Escuela de Computación
Administración de Tecnologías de la Información
Curso TI 3404 – Lenguajes de Programación

En el siguiente documento se presenta el desarrollo y la toma de decisiones para el cuarto proyecto programado, elaborado por los estudiantes Yader Morales, María Mercedes Escalante y Frank Brenes.

23 de Noviembre de 2012

Índice

Introducción

Para el desarrollo de la tarea programada, se utilizaron los conceptos del paradigma orientado a objetos, el cual se distingue por ser llamada una “filosofía”.

La programación orientada a objetos es una metodología que tiene como elemento fundamental el objeto. Permite comprender la representación de este mundo mediante la identificación de los objetos que constituyen el problema, su organización y la representación de sus responsabilidades.

El paradigma orientado a objetos se basa en los tres métodos de organización que utilizamos desde nuestra infancia y en los que basamos todo nuestro pensamiento:

- La diferencia entre un objeto y sus atributos.
- La diferencia entre un objeto y sus componentes.
- La formación y distinción entre clases de objetos.

Algunas de las causas que están influyendo considerablemente en el auge de las programaciones orientadas a objetos son:

- La programación orientada a objetos nos permite trabajar de una forma más organizada.
- Nos posibilita la reutilización del código.
- Nos permite ampliar y utilizar código desarrollado por terceros
- El resultado es un código más limpio, más ordenado y a la larga más fácil de entender y de desarrollar. Hay que tener en cuenta que la mayoría de los objetos comparten una serie de características a nivel sintáctico.

Esta documentación presenta el diseño del programa, decisiones tomadas, estructuras de datos, lógica del programa, instrucciones de compilación, pruebas y resultados de las mismas, así como las referencias técnicas usadas.

Descripción del Programa

El objetivo principal del cuarto proyecto programado es aplicar los conocimientos adquiridos del paradigma de orientación a objetos con una aplicación web que realice las búsquedas de fotos que se encuentren en algunos servicios de publicación de las mismas tales como Flickr o Instagram.

Dentro de los conceptos de Orientación a Objetos que se podrían utilizar para el correcto desarrollo de la tarea programada están los principios básicos de abstracción, modularidad, encapsulamiento, jerarquía, además de las recomendaciones de diseño que fueron vistas en clase.

Por lo que la aplicación deberá tener un formulario de búsqueda, en el cual los usuarios puedan especificar un criterio de selección de fotografías, una vez que el usuario ingresa lo que desea buscar la aplicación, esta se deberá usar el API de búsqueda del servicio de fotos utilizado que puede ser tanto Instagram como Flickr, y obtener las fotos que concuerden con el criterio de búsqueda. El número de fotografías que se obtenga por cada búsqueda podrá ser configurable, es decir el usuario seleccionará la cantidad de resultados que desea que la búsqueda le devuelva, esto puede ser configurable desde la misma aplicación, o desde un archivo de configuración del lado del servidor.

De igual manera, se puede agregar la capacidad a la aplicación web de conectarse con Twitter para realizar tweets de los resultados de las búsquedas.

Diseño del Programa

Para el diseño del programa se decidió conectar con el API de flickr utilizando flickraw, el cual es una librería que permite el fácil acceso al API de Flickr.

La tarea programada consiste de una clase llamada BusquedaFlickr, esta clase es la que se encarga de inicializar las variables de

@Titulo = se encargará de almacenar el título de la fotografía.

@Nombre_usuario = se encargará de almacenar el usuario de la fotografía.

@Url = se encargará de almacenar el url de la fotografía.

@Urls = se encargará de almacenar el urls de la fotografía.

Lo cual es posible gracias al método de initialize presente en la orientación de objetos.

La tarea programada cuenta con los siguientes métodos

- def informacion (tag, num): Este método es la función que llama al API de flickr mediante la librería de Flickraw. La conexión se encuentra presente en esta línea de código

  busqueda_fotos = flickr.photos.search(:tags =>tag, :per_page => num, :page => 1)

  Dentro de la función que está asociada a la variable búsqueda_fotos tenemos que se realizará una búsqueda de las fotos públicas que se encuentren en el servicio de Flickr, buscando por medio de la etiqueta que especifique el usuario, de igual manera se encuentra el atributo :per_page el cual nos permite elegir la cantidad de fotografías a mostrar. El resultado de efectuar la búsqueda de fotos por este método nos devolverá una lista, de todas las fotografías que cumplan con la etiqueta del usuario.

  Una vez que esta función retorna la lista se realiza un ciclo para convertir cada posición de esa lista en un objeto de la clase BusquedaFlickr, en la que se le asigna el título, el usuario y los debidos urls a las variables de clase mencionadas anteriormente. Además de esto, se crea una nueva variable global denominada $lista, la cual contendrá los nuevos objetos creados. Por último el método información llamará al método siguiente (mostrar).

- def mostrar (num): Es el método encargado de crear las variables que serán remplazadas dentro de cada uno de los contenedores dentro del html. En la tarea programada, el html solo cuenta con seis contenedores por lo que solo puede mostrar seis imágenes. Se implementó un botón de siguiente que trabaja, llamando a la misma función, con la lista de resultados creada en el método información. El método mostrar cuenta con los casos de que el usuario quiera menos de seis imágenes a mostrar y en el caso de que requiera más imágenes. Por lo que en cada caso se toman los objetos de la lista, se convierten en variables y se remplazan en el html que se encuentra en Resultado.erb, específicamente en esta sección de código:

```
<img src= <%= $img1 %> height="200px" width="200px">
<p> Titulo: <%= $tit1 %><br>Autor: <%= $aut1 %></br>
```

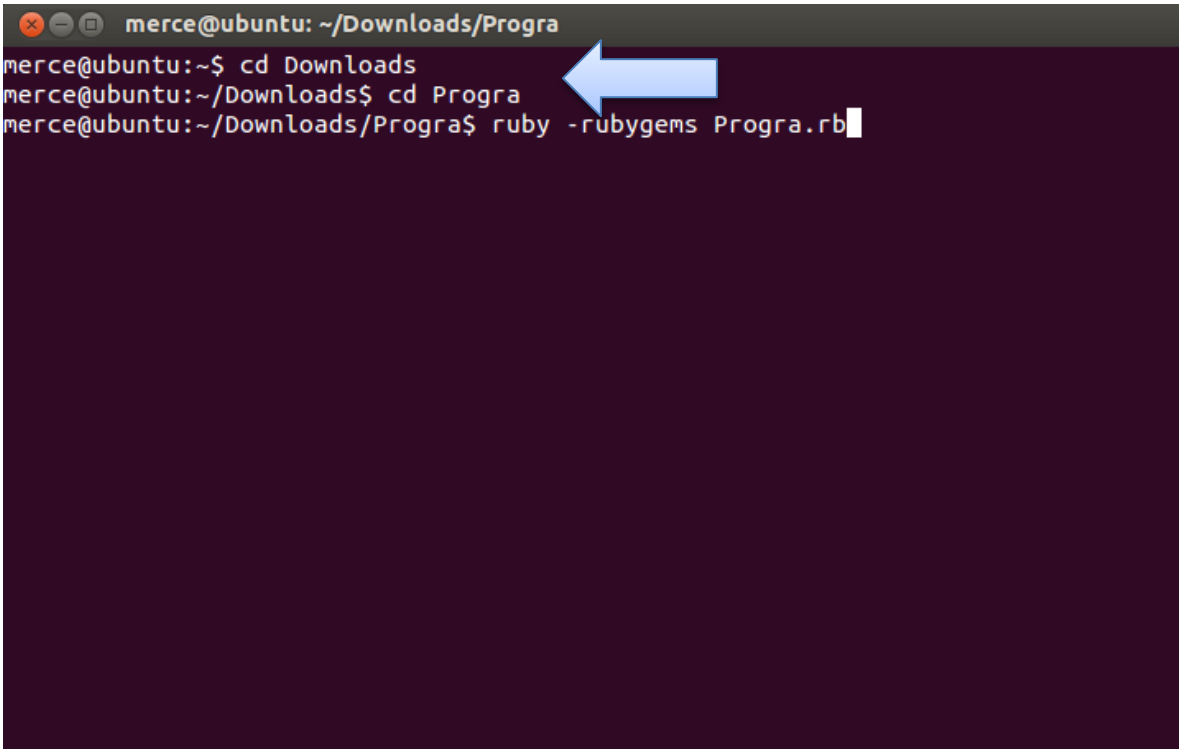
- Como el usuario es posible que requiera más de seis imágenes fue necesario crear los botones de siguiente y anterior, los cuales funcionan mediante las llamadas a toda la tarea programada sin embargo la variable de $Pagina (que funciona como contador dentro de la lista) se aumenta o se disminuye de acuerdo al caso. Además de esto, estos métodos que aumentan o disminuyen a $Pagina para trabajar sobre la lista se encuentran dentro de los métodos post, que corresponden a Sinatra, y son llamados desde los botones que se encuentran en Resultado.erb de la siguiente manera:

```
<CENTER><div id="Siguiente">
    <P>
        <form  action="/siguiente2" method="post">
        <input type="submit" value="Siguiente">
    </P>
        </form>
```

El form_action es lo que llama al método de Sinatra definido de la siguiente manera dentro de la Progra.rb **post '/siguiente2' do** es de esa manera que se pueden desplegar todas las imágenes que desea un usuario.

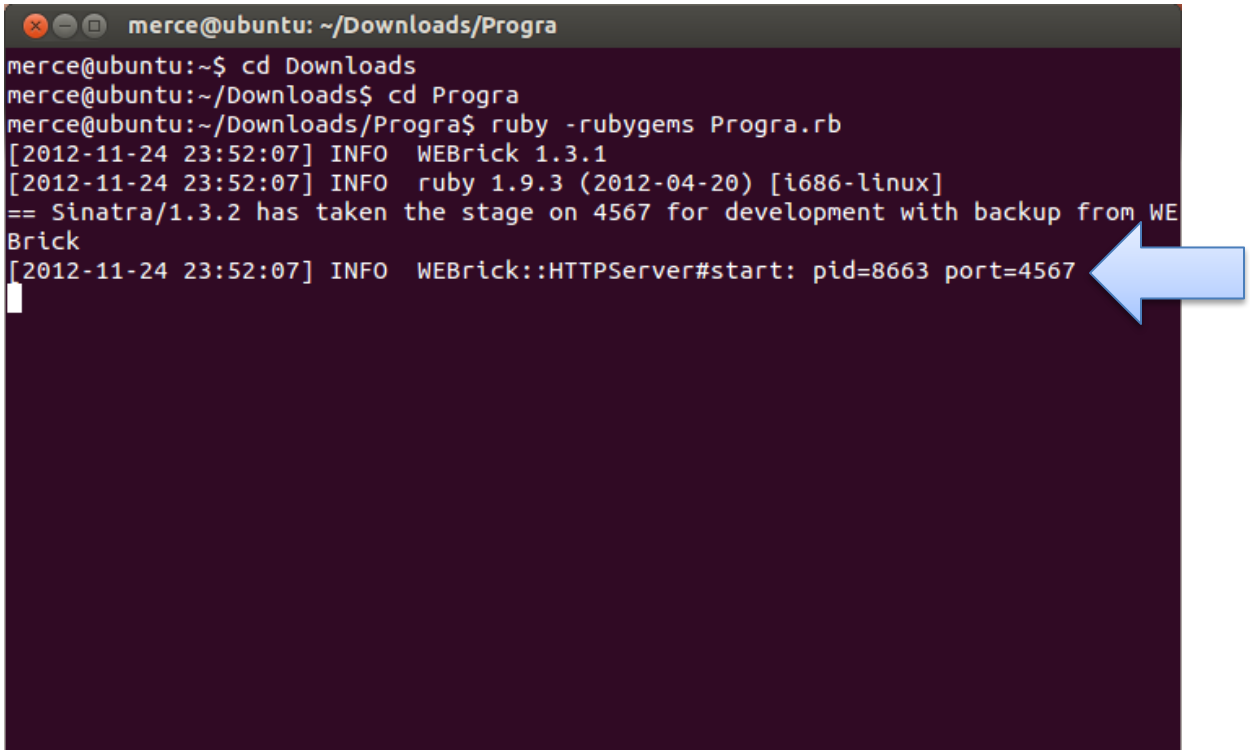
Manual de Usuario

Para iniciar el framework (Sinatra) que se utilizó para la tarea programada es necesario llamar desde consola el programa, de la siguiente forma:



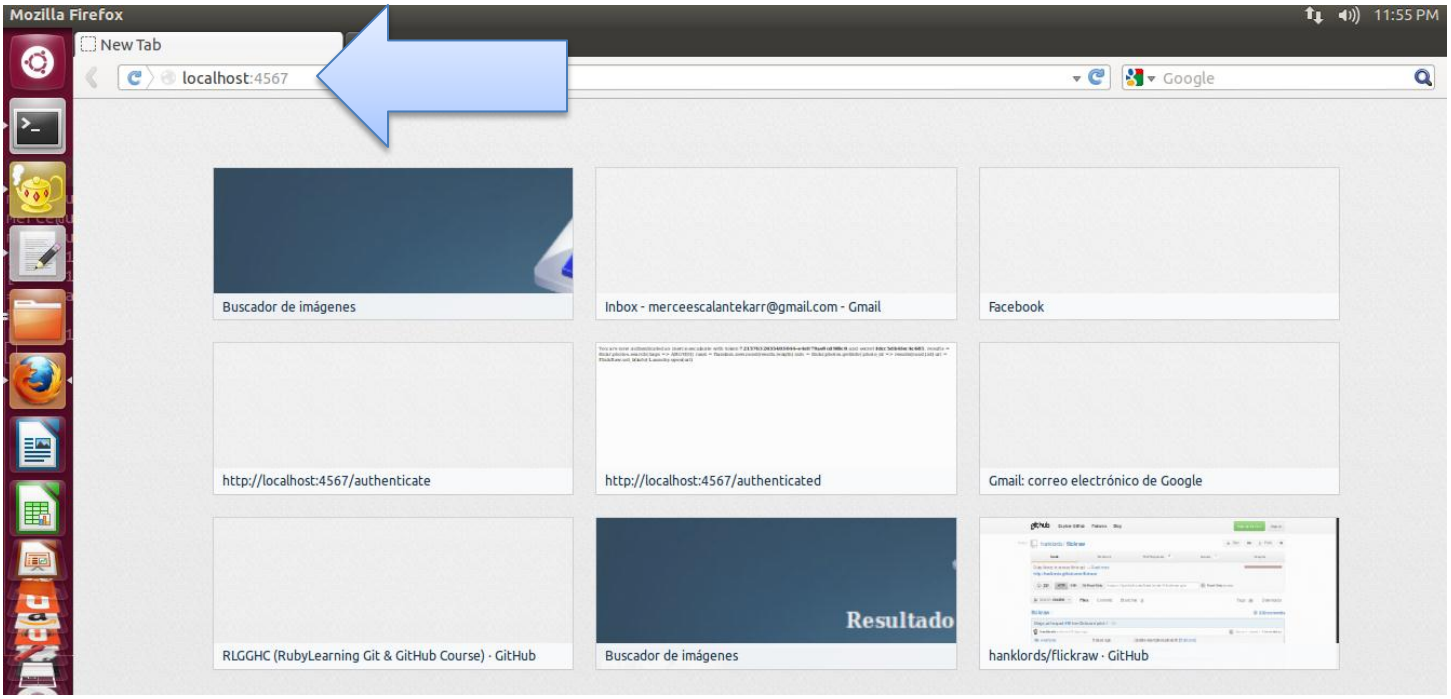
Cabe destacar que no en todas las computadoras es necesario poner –rubygems. Además el usuario debe asegurarse de estar en el directorio donde se encuentra el .rb (extensión de Ruby) o dar la dirección correcta.

Una vez ejecutada la aplicación debería aparecer un mensaje en la consola, similar al que se muestra a continuación.



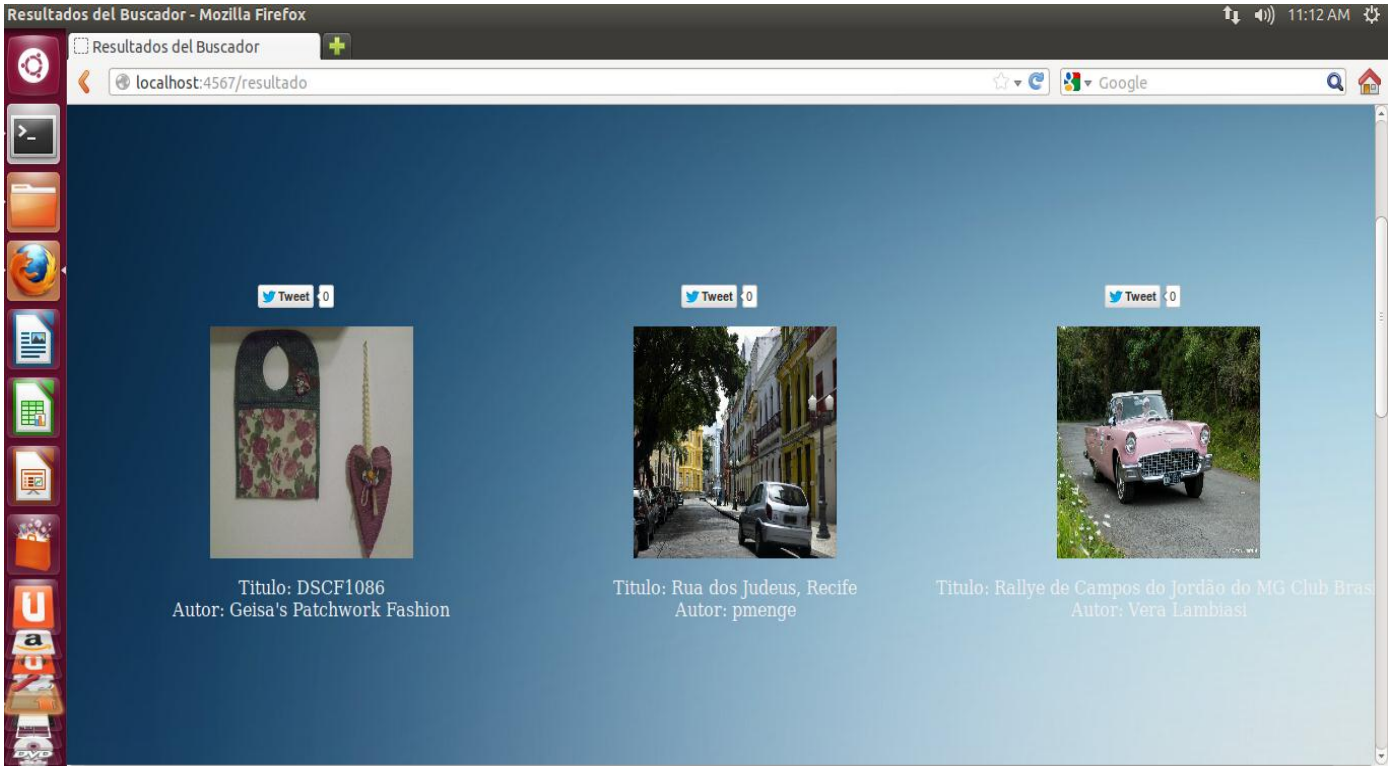
En la imagen anterior se muestra con una flecha donde se levanta Sinatra y de igual manera el puerto que el framework selecciona para levantar la aplicación.

Una vez que aparece el mensaje en la consola de Ubuntu, es necesario ingresar al navegador específicamente a la dirección localhost:4567.

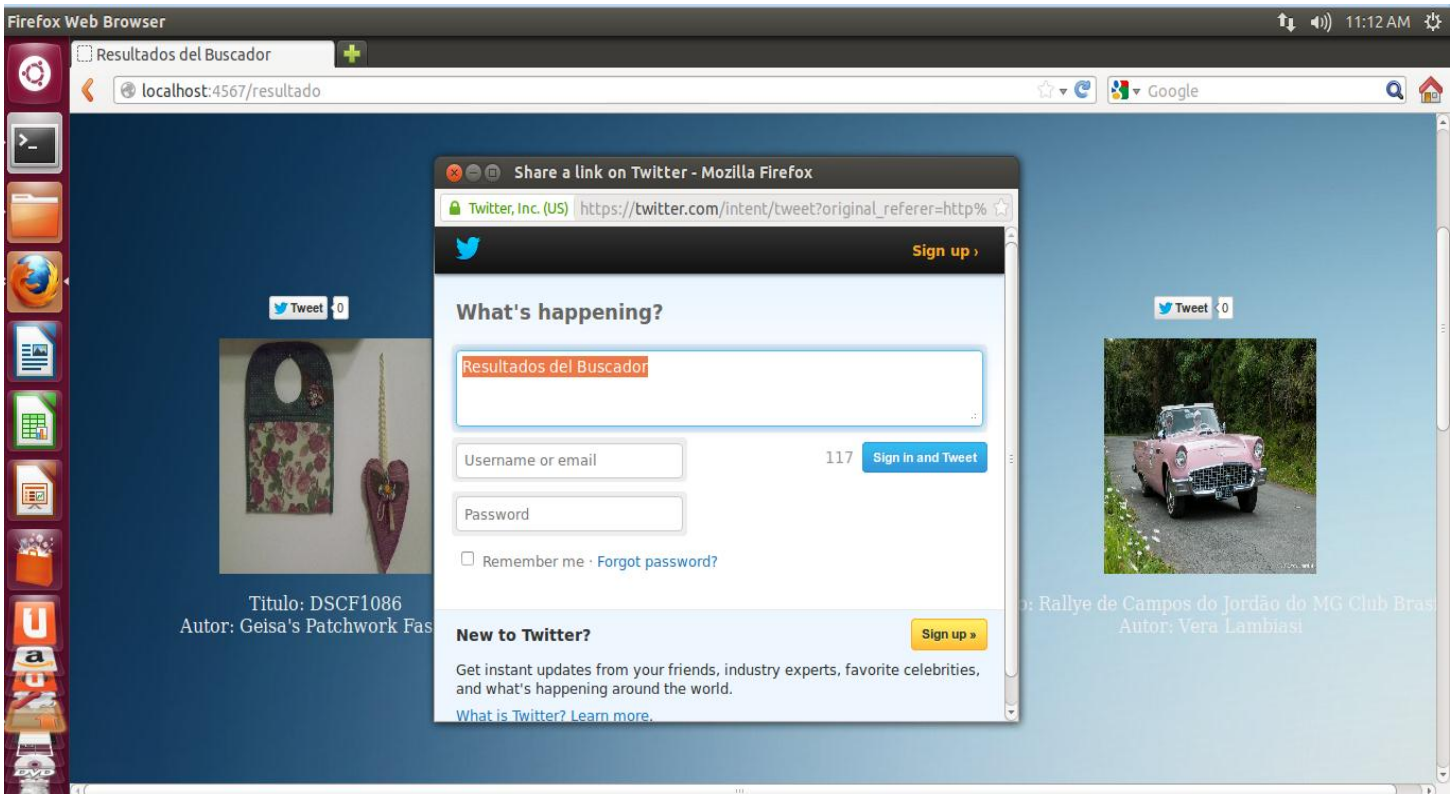


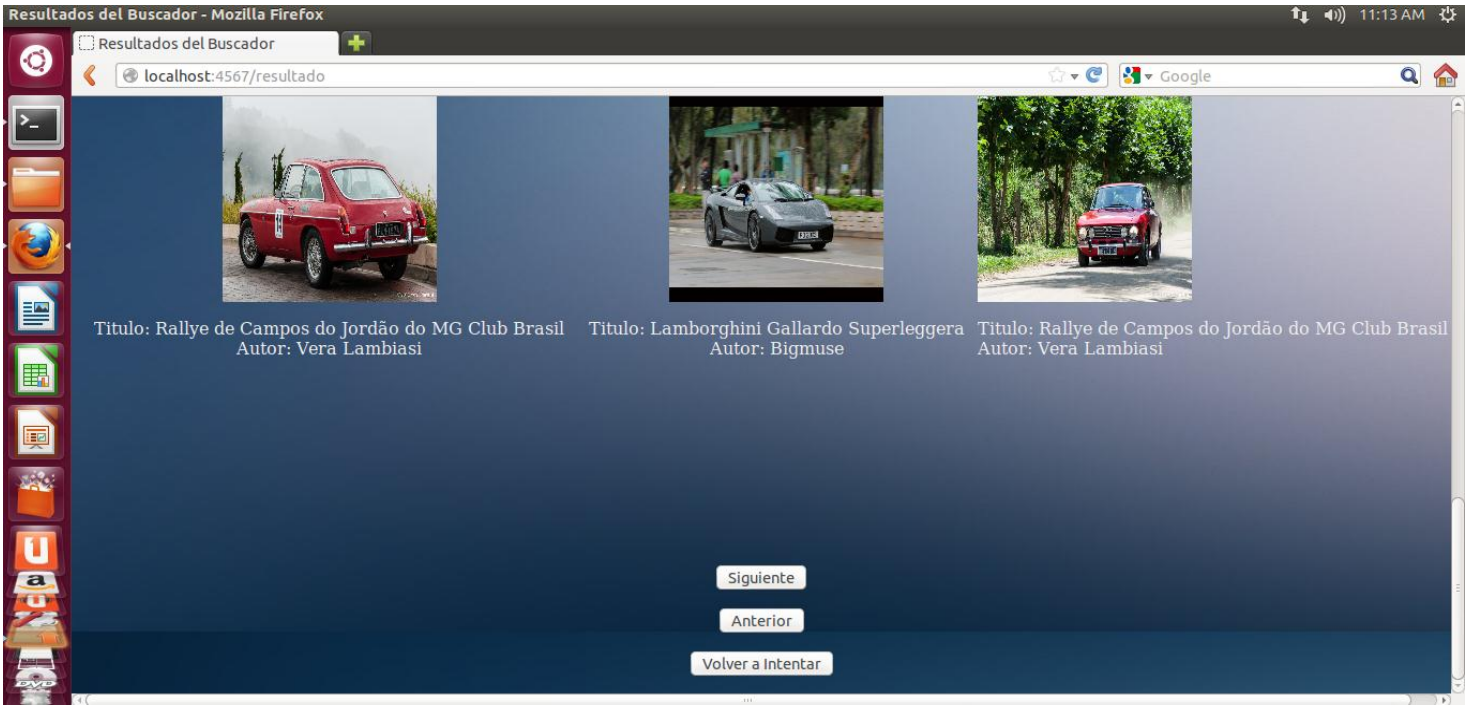
Una vez ejecutado el comando se mostrará la interfaz de la aplicación, tal y como se presenta en la siguiente imagen.

Una vez que el usuario selecciona la cantidad de resultados que desea, y la etiqueta y presiona el botón de buscar se le desplegará la siguiente pantalla al usuario.



Se le presenta la opción de realizar un tweet con los resultados de la búsqueda. Si el usuario presiona el botón de Tweet se le presentará la siguiente ventana:





Si el usuario llega al final de la pantalla, se le presenta la opción de darle al botón siguiente, anterior o volver a intentar el cual lo redireccionará la pantalla de búsqueda.

Análisis de Resultados

El proyecto programado fue finalizado con las siguientes funcionalidades implementadas:

- Utilización de los conceptos de Programación Orientada a Objetos en la definición de la clase BusquedaFlickr.
- Conectarse con el servicio de API del servicio de fotografías Flickr.
- Utilización de Sinatra para la aplicación web.
- Búsqueda de fotos por medio de la aplicación web.
- Permitir al usuario seleccionar la cantidad de resultados que desea para su búsqueda.
- De igual manera la tarea programada se conecta con Twitter y se puede hacer el respectivo tweet de los resultados de las fotos.

Proceso de Instalación

Para la instalación de Ruby se utiliza el comando:

```
% sudo apt-get install ruby irb rdoc
```

Para la instalación de Sinatra es necesario:

```
sudo apt-get install ruby-sinatra
```

Para la instalación de Flickraw

```
gem install json (Necesaria en caso de tener ruby 1.8)
gem install flickraw
```

Conclusión

Luego de desarrollar el cuarto proyecto programado dentro de un paradigma muy diferente a los estudiados anteriormente dentro del curso (los cuales son el paradigma imperativo, lógico, y el funcional), podemos rescatar la importancia de comprender las distintas aplicaciones dentro de los lenguajes de programación que se encuentran bajo el paradigma de orientación a objetos. De igual manera las características que diferencian este paradigma de los otros, además de la aplicación de los conceptos básicos de este lenguaje tales como la herencia, y el polimorfismo.

Durante el desarrollo de la tarea programada fomentamos habilidades claves para ser programadores exitosos, tales como la investigación, la cual fomentamos con el estudio de la plataforma necesaria para manejar la conexión al API de Flickr (el cual fue el elegido para el buscador de fotos de la tarea programada), de igual manera fue necesario implementar los conocimientos de investigación para desarrollar la tarea sobre el framework de Sinatra y la estructura de los archivos HTML utilizados junto con el uso de la librería erb para manejar la interfaz gráfica de la aplicación.

Es importante recalcar, que con el desarrollo de esta tarea no solo aprendimos sobre distintos puntos de vista para resolver un mismo problema; si no también reforzamos el trabajo en equipo, el desarrollo de algoritmos en conjunto, conocimientos de cada uno de los integrantes y lo más importante nos enseñó a motivarnos entre nosotros para lograr concluir una tarea en común.

Bibliografía

Hanklords/flickraw · GitHub (s. f.). Recuperado el de 16 Noviembre del 2012, de https://github.com/hanklords/flickraw

Class FlickRaw::Flickr::Groups - RDoc Documentation (s. f.). Recuperado el 16 de Noviembre del 2012, de http://hanklords.github.com/flickraw/FlickRaw/Flickr/Groups.html#method-i-search

Flickr Services: Flickr API: flickr.photos.search (s. f.). Recuperado el 19 de Noviembre del 2012, de http://www.flickr.com/services/api/flickr.photos.search.html

Flickr Services (s. f.). Recuperado el 19 de Noviembre del 2012, de http://www.flickr.com/services/api/

Desarrollo en NET: DevLab: Introducción a Sinatra (s. f.). Recuperado el 12 de Noviembre del 2012, de http://mario-chavez.blogspot.com/2009/10/devlab-introduccion-sinatra.html

Class: ERB (Ruby 1.9.3) (s. f.). Recuperado el 26 de Noviembre del 2012, de http://ruby-doc.org/stdlib-1.9.3/libdoc/erb/rdoc/ERB.html

Sinatra: README (s. f.). Recuperado el 17 de Noviembre del 2012, de http://www.sinatrarb.com/intro.html

How WFT install ruby-sinatra ? (s. f.). Recuperado el 20 de Noviembre del 2012, de http://www.imperiosweb.com/howto/how-to-install/ruby-sinatra.html

Valibuk.net » Access Flickr with Ruby and Flickraw (s. f.). Recuperado el 19 de Noviembre del 2012, de http://www.valibuk.net/2010/03/access-flickr-with-ruby-and-flickraw/