

Tarea programada

Orientacion a Objetos - Ruby

Fernanda Fernández
Melissa Gutiérrez
Jean Carlo Alfaro

6/15/2012

Table of Contents

Descripción del programa2

Librerías Usadas3

Diseño del programa3

Decisiones de tomadas.....6

Conexión con Twitter, manejo de OAuth7

Análisis de resultado9

Manual de Usuario9

Conclusiones12

Descripción del programa

Se debe desarrollar un programa que permita tomar cierto contenido, seleccionado por el usuario, de la página Bandcam y dicho contenido sea publicado en una cuenta de Twitter. El programa se debe de desarrollar en el lenguaje Ruby, y poder ejecutarse en el sistema operativo Linux.

El objetivo es que se puede promover el campo de la música en una red social, para que llegue a la mayor cantidad de personas posibles, ya que algunas bandas presentan dificultad para divulgar su música, de esta forma, los seguidores de dicha cuenta puedan ver la información de las bandas.

El programa debe permitir al usuario digitar una etiqueta (puede ser por género musical o por ubicación) para posteriormente hacer la búsqueda correspondiente en la página de Bandcamp, y obtener de los primeros 10 resultados que aparezcan en la página, la siguiente información:

- Nombre del grupo
- Nombre del álbum
- Gratuito o no
- Url de bandcamp del grupo

Después de haber obtenido los resultados de Bandcamp, para cada resultado obtenido, se debe publicar un Twitts que contiene dicha información. Para poder publicar contenido en Twitter, deberán autenticarse con el API, para lo cual deberán hacer uso del mecanismo de autenticación OAuth.

Librerías Usadas

Estas librerías debes instalarlas desde consola con los comando siguientes:

- `sudo apt-get install ruby`: instalar ruby
- `sudo apt-get install rubygems`: libreria rubygems para correr aplicaciones gems
- `sudo gems install rubygems-update`: actualiza rubygems
- `sudo gem install twitter4r`: instala un elemento que sirve para rubygems
- `sudo gem-install twitter` :libreria para conectar ruby con twitter
- `sudo gem install hpricot`: libreria para navegar en un html/xml

Diseño del programa

El programa es diseñado según los requerimientos de las 4 consultas base. Para cada una de ellas se realizaron diferentes funciones que permiten obtener los 10 principales datos solicitados por el usuario, para ello se requirió de estructuración de una clase principal Cancion. Esta adapta cuatro atributos importantes, las cuales son: url, álbum, autor y precio; en ellos se almacenaran los datos principales requeridos.

Veamos la definición de la clase Cancion:

```
class Cancion
  attr_accessor:url
  attr_accessor:album
  attr_accessor:autor
  attr_accessor:precio

  def seturl(nuevo_url)
    url = nuevo_url
    puts "url -> "+ url
  end

  def setalbum(nuevo_album)
    album = nuevo_album
    puts "Album -> "+ album
  end

  def setautor(nuevo_autor)
    autor = nuevo_autor
    puts "Autor -> "+ autor
  end

  def setprecio(nuevo_precio)
    precio = nuevo_precio
    puts "Precio -> "+ precio
  end
end
```

Para obtener los datos de Bandcamp utilizamos un algoritmo que utiliza los url de la página web, y los añade a un array de los cuales vamos a ir separándolos para obtener los datos necesario solicitados en la especificación del tarea programada.

```
class Recorrido
```

```
  def initialize(url)
    @url = url;
    @hp = Hpricot(open(@url))
  end

  def recorre(link)
    array = []
    bandera = 0
    cont = 0
    open(link) do |f|
      f.each do |line|
        if line == "popularity\n"
          bandera = 1
        end
        if line == ".pager {\n"
          bandera = 2
        end
        if bandera == 1
          x = recorre_aux(line)
          if x != "vacio" and cont < 11
            array << [x]
            cont = cont + 1
          end
        end
      end
    end
    return array
  end
end
```

Nosotros implemetamos la búsqueda por link, básicamente el código anterior recorre un link , desde el código fuente de la pagina de bandcamp, posteriormente sobre el se encuentra un if el cual permite que si encuentra:

```
  if line == "popularity\n"
    bandera = 1
```

que realiza que a la hora de encontrar la palabra “popularity” es cuando encontramos los url que necesitamos, por lo que se aumenta el contador llamado bandera para que pare la búsqueda.

Luego en el mismo código llamamos a la función recorrido_line para buscar por línea la información requerida.

```
def recorrido_line(line)
  bandera1=0
  line.each_byte do |x|
    if x == 97
      if bandera1 == 1
        return recorrido_byte(line)
        bandera1 = 0;
      end
    else
      bandera1 = 0
    end
    if x == 60
      bandera1 = 1
    end
  end
  return "vacio"
end
```

En este código nos llama a la función recorrido_byte el cual concatena el url definido con lo que el usuario decide buscar ya sea género o ciudad.

```
def recorrido_byte(line)
  bandera1 = 0
  bandera2 = 0
  link = ""
  line.each_byte do |x|
    if x == 34 and bandera1 == 1
      return link
    end
    if bandera1 == 1
      link=link.concat(x.chr)
    end
    if x == 34
      bandera1 = 1
    end
  end
end
```

Decisiones de tomadas

En este algoritmo extraemos los datos del costo de las canciones ya sea Free= gratuito o Pay= pagado esto con el fin de que el usuario obtenga los necesario para el usuario.

```
def costo(url)
  @link = url;
  @hp2 = Hpricot(open(@link))
  rating_text = (@hp2/"h4.compound-button").inner_text
  y="\n      \n      \n      Free Download\n      \n      \n      \n      "

  if rating_text== y
    return "Free"
  else
    return "Pay"
  end
end
```

Con este otro algoritmo podemos extraer los nombres de el álbum y el grupo, desglosándolos

```
def AutorAlbum(url)
  url = url;
  hp = Hpricot(open(url))
  grupoAlbum = hp.at("meta[@name='title']")["content"]
  desglosa(grupoAlbum)

end

def desglosa(grupoAlbum)
  grupoAlbum = grupoAlbum.split(', by ')
  grupoAlbum

end
```

Luego se realiza un algoritmo el cual posee un objeto que va a contener el URL, AUTOR, si es PAY o FREE entre otros.

```
def datos(url)
  x = AutorAlbum(url)
  nueva_cancion = Cancion.new
  nueva_cancion.seturl(url)
  nueva_cancion.setalbum(x[0])
  nueva_cancion.setautor(x[1])
  nueva_cancion.setprecio(costo(url))

end
```

Al final del algoritmos antes mencionada se le agregan 2 lineas de codigo las cuales permiten al usuario tweetear el contenido deseado.

```
tw = Tweet.new
tw.tweetear(url, x[0], x[1], costo(url))
```

Finalmente el usuario podra observar un top10 en la cual se encuentra las 10 primeras canciones segun los que el usuario ingreso que deseaba buscar.

```
def top10(url)
  lista=recorre(url)
  for i in 1..10
    datos(lista[i][0])
  end
end
end
```

Conexión con Twitter, manejo de OAuth

```
class Autenticacion
  def initialize()
    #La aplicación está registrada con los siguientes datos
    @token="TY7zz27iEF1CWY0sjK1cA"
    @secret ="WV0J7FynkbpDAgsqSEhFcGK4lNqm4BnKHTTZWVfCs"
  end
end
```

En esta primera funcion del codigo digitamos los @token y @secret de la aplicación creada en twitter y básicamente realiza la autenticación de la misma.

Luego realizamos la conexión

```
def conection
```

Empezamos a desarrollar el cliente para oauth

```
cliente=OAuth::Consumer.new(
  @token,
  @secret,
  {
    :site=>"http://twitter.com",
    :request_token_url=>"https://api.twitter.com/oauth/request_token",
    :access_token_url =>"https://api.twitter.com/oauth/access_token",
    :authorize_url    =>"https://api.twitter.com/oauth/authorize"
  }
)
```

En esta se solicita al api el token y el secret del usuario

```
request_token = cliente.get_request_token
token2 = request_token.token
secret2 = request_token.secret
```

Aquí se abre el navegador predeterminado del sistema de la autorización

```
direccion = cliente.authorize_url + "?oauth_token=" + token2
puts "Abriendo en el navegador: "+direccion
system('start '+direccion)
#solicita el pin brindado por twitter
print "Clic al link anterior e ingrese el pin que aparese en la pagina del Tweeter de su navegador:"
pin = gets.chomp
puts
```


Luego se autentica al usuario con los datos brindados

```
begin
  OAuth::RequestToken.new(cliente, token2, secret2)
  access_token=request_token.get_access_token(:oauth_verifier => pin)
  Twitter.configure do |config|
    config.consumer_key = @token
    config.consumer_secret = @secret
    config.oauth_token = access_token.token
    config.oauth_token_secret = access_token.secret
  end
  $client = Twitter::Client.new
  $client.verify_credentials
  puts "Autenticado Correctamente"

rescue Twitter::Unauthorized
  puts "Error de Autorizacion"
end
end
end
```

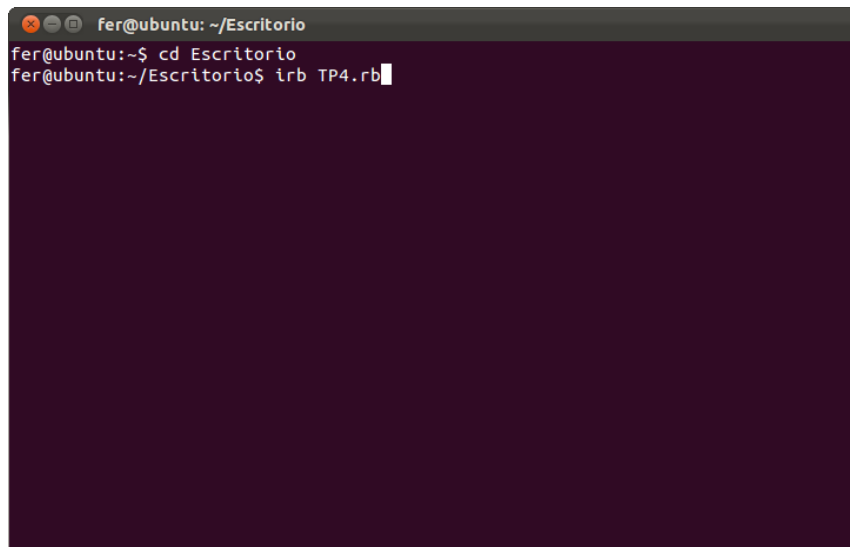
Analisis de resultado

Se logró finalizar con todos los aspectos requeridos por el usuario final, es decir, obtener y clasificar el contenido de Bandcamp, conectarse al API de Twitter y poder publicar tweets desde un programa en Ruby y manejo de autenticación por medio de OAuth.

Manual de Usuario

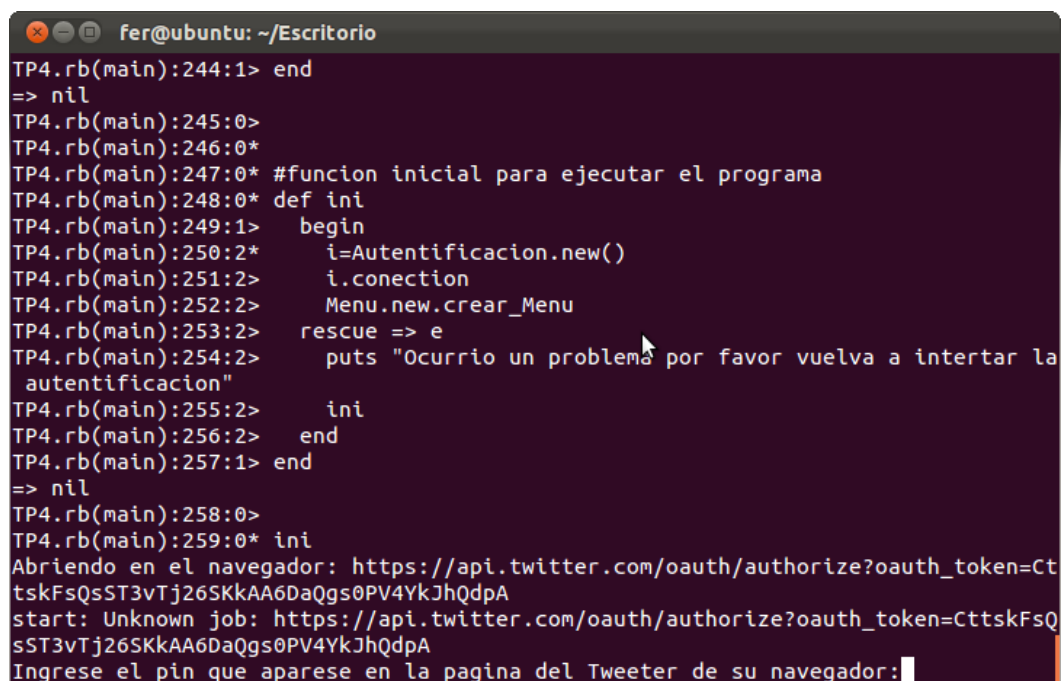
Primeramente para correr esta aplicacion debe de instalar las librerias antes descritas para el funcionamiento total de la aplicación.

Luego debe correr la aplicación de la siguiente manera:



```
fer@ubuntu: ~/Escritorio
fer@ubuntu:~$ cd Escritorio
fer@ubuntu:~/Escritorio$ irb TP4.rb
```

Presiona Enter y se empieza a ejecutar la aplicación, en pantalla se le mostrara algo similar:



```
fer@ubuntu: ~/Escritorio
TP4.rb(main):244:1> end
=> nil
TP4.rb(main):245:0>
TP4.rb(main):246:0*
TP4.rb(main):247:0* #funcion inicial para ejecutar el programa
TP4.rb(main):248:0* def ini
TP4.rb(main):249:1>   begin
TP4.rb(main):250:2*     i=Autenticacion.new()
TP4.rb(main):251:2>     i.conection
TP4.rb(main):252:2>     Menu.new.crear_Menu
TP4.rb(main):253:2> rescue => e
TP4.rb(main):254:2>   puts "Ocurrio un problema por favor vuelva a intertar la
  autenticacion"
TP4.rb(main):255:2>   ini
TP4.rb(main):256:2> end
TP4.rb(main):257:1> end
=> nil
TP4.rb(main):258:0>
TP4.rb(main):259:0* ini
Abriendo en el navegador: https://api.twitter.com/oauth/authorize?oauth_token=Ct
tskFsQsST3vTj26SKkAA6DaQgs0PV4YkJhQdpA
start: Unknown job: https://api.twitter.com/oauth/authorize?oauth_token=CttskFsQ
sST3vTj26SKkAA6DaQgs0PV4YkJhQdpA
Ingrese el pin que aparese en la pagina del Tweeter de su navegador:
```

Restaurar sesión

Twitter / Autorizar una apli...

twitter.com https://api.twitter.com/oauth/authorize?oauth_token=nTuyd68dgmWYTqfOTr2KpyllONTfewT5ozm257j

Authorize Progra-ruby to use your account?

Esta aplicación **será capaz de:**

- Leer Tweets de tu cronología.
- Ver a quién sigues y seguir a nuevas personas.
- Actualizar tu perfil.
- Publicar Tweets por ti.

Nombre de usuario o correo electrónico

Contraseña

☐ Recuerda mis datos · [¿Olvidaste tu contraseña?](#)

[Autorizar la aplicación](#) [No, gracias](#)

Esta aplicación **no tendrá capacidad para:**

- Acceder a tus mensajes directos.
- Ver tu contraseña de Twitter.

Progra-ruby
www.facebook.com/fernanda.fde...
Lenguajes de programacion/Ruby

En cualquier momento, puedes revocar el acceso a cualquier aplicación desde la [lista de aplicaciones](#) de tu página de configuración.

Al autorizar una aplicación, continuarás operando bajo las [Condiciones de Servicio de Twitter](#). En concreto, algunos datos de uso serán compartidos con Twitter. Para más información, consulta nuestra [Política de Privacidad](#).

Posteriormente debes darle clic alguno de los links que aparecen, ellos te van a vincular a una página del twitter el cual les va a dar un pin para ejecutar la aplicación.

Restaurar sesión

Twitter / Autorizar una apli...

twitter.com https://api.twitter.com/oauth/authorize

¡Has concedido acceso a Progra-ruby!

A continuación, regresa a Progra-ruby e introduce este PIN para completar el proceso de autorización:

5377197

[← Ir a Twitter](#) [Ir a la página de inicio de Progra-ruby](#)

En cualquier momento, puedes revocar el acceso a cualquier aplicación desde la [lista de aplicaciones](#) de tu página de configuración.

Al autorizar una aplicación, continuarás operando bajo las [Condiciones de Servicio de Twitter](#). En concreto, algunos datos de uso serán compartidos con Twitter. Para más información, consulta nuestra [Política de Privacidad](#).

NOTA: Debes logearte antes.

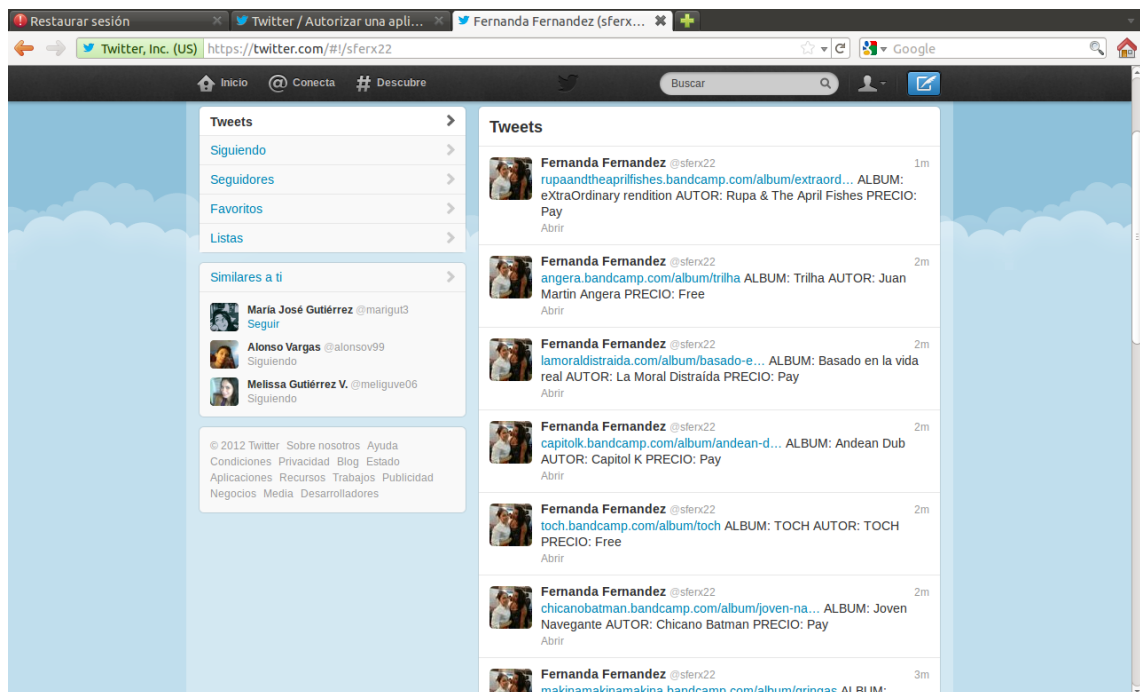
Luego se le presenta al usuario un menú en el cual el digita el genero o la ciudad en la cual quiere desarrollar su búsqueda.

```
fer@ubuntu: ~/Escritorio
Autenticado Correctamente
Bienvenid@ a AFM
-----Pasos-----
1. Digite el género o ubicación deseada
rock
-----
url =====> http://jimguthrie.bandcamp.com/album/indie-game-the-movie-soundtrack
Album =====> Indie Game: The Movie (Soundtrack)
Autor =====> Jim Guthrie

Precio =====> no gratuito

-----
url =====> http://dbsoundworks.bandcamp.com/album/the-binding-of-isaac-2
Album =====> The Binding of Isaac
```

Y así se muestran los 10 resultados solicitados, los cual son tweeteados.



Conclusiones

El curso de Lenguajes de Programación y la tarea programada permitió conocer de una forma más profunda sobre el lenguaje de Ruby, como estructura, sintaxis, manejo de las clases, etc. Además la importancia que tiene el desarrollo de programas mediante lenguajes orientado a objeto, debido a que se ha incrementado el desarrollo de programas en este tipo de paradigma.

Los programas OO permiten obtener un código más limpio, más ordenado y 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.

Para el desarrollo del proyecto programado es necesario tomar en cuenta ciertos elementos para lograr y cumplir con los requerimientos deseados.

* Primero se debe comprender en su totalidad los requerimientos deseados, investigar y realizar una previa preparación del lenguaje que se solicita la tarea. Posteriormente se debe identificar elementos como el objetivo final, dividir en etapas la tarea y que realizará cada integrante del equipo de trabajo y que se realizará en conjunto. Esto permite una buena integración y ordinación de la totalidad de la tarea.

* Es importante realizar una investigación profunda y exhaustiva de tema desconocidos por el equipo de trabajo y que son fundamentales para el desarrollo del proyecto. Algunos aspectos que fueron importantes investigar son:

- Obtener y clasificar el contenido de Bandcamp.
- Conectarse al API de Twitter y poder publicar tweets desde un programa en Ruby.
- Manejo de autenticación por medio de OAuth.

* Se debe ir monitoreando el desarrollo y el avance del proyecto, esto es muy importante porque permite detectar errores o un incorrecto entendimiento de requerimientos, en el mejor tiempo posible.