

Alejandro Pelaéz Roldán  
200727000010

## Parcial 2

### Guía de uso

Escogí el segundo método de integración, que es el que cuando se recibe una petición se redirige esta hacia todas las bibliotecas.

Para esto se implementarán dos sistemas. El primero se llama **yafdl**, este es el encargado de hablar con los clientes mediante el browser, consta de dos simples interfaces, una de búsqueda y una de resultados. En la interfaz de búsqueda simplemente se ingresan los parametros deseados y se da click en ok, luego se redirecciona a la interfaz de resultados donde se visualiza todo en una tabla.

**Yafdl** cuenta con un archivo de configuración el cual está en la carpeta */public/servers/servers.xml*. Este archivo es simplemente una lista de los servidores BiDi disponibles, se pueden añadir nuevos servidores simplemente añadiendo nodos, pero es importante asegurarse que el campo url esta bien dado porque de lo contrario la conexión fallará.

El segundo sistema **BiDi\_SRW** este es el que contiene la información (tanto los metadatos como los datos), los metadatos van en la carpeta *public/metadata* y los datos en *public/data* se pueden añadir metadatos nuevos lo importante es que apunten a un dato real que exista en la carpeta data, y tambien es importante que el campo url sea completo, es decir que incluya el protocolo (http, ftp) host, puerto y uri.

Este sistema expone un servicio que se llama query, se accesa mediante la url

“*http://host:puerto/metadata*”, los parametros del query pueden ser pasados o por el query string (url) o como parametros del request http, y se debe acceder a este servicio por GET (para seguir la convención REST). Este servicio simplemente regresa un xml con nodo raiz *<metadata>*, adentro de este esta toda la información recuperada que hizo match con los parámetros pasados.

Los parámetros pueden ser *field* este es por que campo del metadato buscar, y *term* la palabra a buscar.

Para generar más bidis simplemente se copia la carpeta *bid\_i\_SRW*, y hay que tener en cuenta en modificar los metadatos con el host y el puerto correspondiente.

### Instalación y ejecución

Primero, se necesita tener Ruby 1.9.2 instalado. Para esto, se recomienda instalar RVM.

```
$ bash < <(curl -s https://rvm.beginrescueend.com/install/rvm)
```

Una vez instalado RVM, se debe instalar Ruby 1.9.2:

```
rvm install 1.9.2
```

Una vez instalada Ruby 1.9.2, se deben instalar las dependencias:

```
rvm 1.9.2  
gem install bundler
```

Ahora sí, en la entrega se tienen 4 carpetas:

- yafdl
- bidi\_SRW
- bidi\_SRW2

Se debe ingresar primero a cada carpeta y correr el comando

```
rvm 1.9.2  
bundle install
```

Con esto ya quedan instaladas todas las dependencias necesarias. Luego se entra a cada bidi(en nuestro caso son solo 2) y en la primera (bidi\_SRW) se ingresa el comando *rails s -p 9002 -e production* para iniciar un servidor en el puerto 9002 (porque eso fue lo que pusimos en los metadatos).

Para bidi\_SRW lo mismo pero el comando sería *rails s -p 9003 -e production para correrlo en el puerto 9003*.

Sólo faltaría por correr el servidor *yafdl*, entonces para esto se ingresa en esa carpeta y se ingresa el comando *rails s* como no se especifico puerto este corre en el 3000.

Para ingresar por medio del browser se ingresa a la dirección “<http://localhost:3000>” esto inmediatamente llevará al ormulario de búsqueda, se ingresan los parámetros deseados y listo.