Diseño de una API y APP en Rails

# Diseño de una API Rails que muestre noticias de sitios web como Mashable, Digg y Reddit y diseño de una APP en Rails que consuma esta API

# **CREACIÓN DE LA API CON RAILS-API:**

Puedes visitar el documento de Rails-Api, para más información y puedas ver la diferencia entre hacer un proyecto con Rails-api y completo en Rails

<https://github.com/rails-api/rails-api>

Crearemos el proyecto con MySQL como Base de Datos:

Ejecutar:→ **rails-api new -d mysql api\_feeds**

## Utilizaremos la gema HTTParty

<https://github.com/jnunemaker/httparty>

Agregar la gema en **app/Gemfile → gem 'httparty'**

Instalarla ejecutando: → **gem install httparty**

En la terminal y dentro de la carpeta ejecutar **→ bundle install**

## Creación y migración de la Base de Datos:

Ejecutar → rake db:create; rake db:migrate

## Creación del modelo: feed.rb

Dentro de la carpeta **app/models**

El archivo va a requerir **“httparty”** y la clase del modelo incluye **HTTParty** y **ActiveModel**:

**require 'httparty'**

**class Feed**

**include ActiveModel::Model**

**include HTTParty**

**.**

**.**

**.**

**end**

## **Creación del** **CONTROLADOR**:

El nombre va en mayúscula y Plural

Ejecutar → **rails g controller Feeds**

## 2 SERVIDORES instalaremos Rails Api Cors:

<http://winn.ws/blog/2014/04/20/rails-api-cors/>

Agregar la gema en **app/Gemfile → gem 'rack-cors', :require => 'rack/cors'**

En el archivo **aplication.rb** ubicado en **app/config/aplication.rb**

Eliminar la línea 25 → **config.active\_record.raise\_in\_transactional\_callbacks = true**

Pegar:

**config.middleware.use Rack::Cors do**

**allow do**

**origins '\*'**

**resource '\*', :headers => :any, :methods => [:get]**

**end**

**end**

En la terminal y dentro de la carpeta ejecutar **→ bundle install**

# **CREACIÓN DE LA APP EN RAILS:**

Nueva APP (consumir API) → https://github.com/rails/rails

**→ rails new -d mysql app\_feeds**

## Para utilizar los 2 servidores instalaremos la siguiente gema (no es necesario):

<https://github.com/macournoyer/thin/>

(A thin and fast web server)

Agregar la gema en **app/Gemfile → gem ‘thin’**

Instalarla ejecutando: **→gem install thin**

En la terminal y dentro de la carpeta ejecutar **→ bundle install**

## Para agilizar los procesos e incrementar velocidad en HTTP:

<https://github.com/drbrain/net-http-persistent>

(Using persistent HTTP connections can dramatically increase the speed of HTTP)

Agregar la gema en **app/Gemfile → gem 'net-http-persistent', '~> 2.9.4'**

Instalarla ejecutando: → **gem install net-http-persistent**

En la terminal y dentro de la carpeta ejecutar **→ bundle install**

## Para mejorar la presentación de los query:

<https://github.com/activerecord-hackery/squeel>

(Squeel lets you write your Active Record queries with fewer strings, and more Ruby)

Agregar la gema en **app/Gemfile → gem 'squeel', '~> 1.2.3'**

Instalarla ejecutando: → **gem install squeel**

En la terminal y dentro de la carpeta ejecutar **→ bundle install**

Ejecutar **→ rails g squeel:initializer**

## Creación del **MODELO**.

El nombre va en **Mayúscula y singular**

Ejecutar →**rails g model Feed**

## Creación del CONTROLADOR:

El nombre va en mayúscula y Plural

Ejecutar → **rails g controller Feeds**

## Creación y Migración de la Base de Datos:

Ejecutar → **rake db:create; rake db:migrate**

## Agregando BOOTSRAP CSS stylesheets

<https://github.com/seyhunak/twitter-bootstrap-rails>

Agregar la gema en **app/Gemfile**

**gem "therubyracer"  
gem "less-rails" #Sprockets (what Rails 3.1 uses for its asset pipeline) supports LESS  
gem "twitter-bootstrap-rails"**

Instalarla ejecutando: **→rails generate bootstrap:install static**

En la terminal y dentro de la carpeta ejecutar **→ bundle install**

**Configurar el archivo aplication.css**

Buscar el archivo **BOOTSTRAP\_AND\_OVERRIDES** ubicado en **app/assets/stylesheets/bootstrap\_and\_overrides**

* 1. Copiar líneas 2 y 6 y pegarlas en el archivo **aplication.css**

**linea 2: → \*= require twitter-bootstrap-static/bootstrap**

**linea 6: → \*= require twitter-bootstrap-static/fontawesome**

* 1. Eliminar archivo **bootstrap\_and\_overrides**
  2. Pegar las 2 líneas anteriores en el archivo **aplication.css** ubicado en: **app/assets/stylesheets/aplication.css** pegarlas luego de la línea 13 que tiene escrito **\*= require\_tree .**

## Generar el LAYOUT (index, mashable, digg, reddit, digg)

Ejecutar → **rails g bootstrap:layout NOMBRE**

Mover el archivo a la carpeta **app/views/feeds**

Configurar route agregando la ruta **root 'feeds#index'**

## Dándole Dinamismo a la App con Javascript

Agregar los archivos **.js** en la ruta: **app/assets/javascripts/**

## Dándole Estilo a la App

El estilo a la app se agregan en el archivo **application.css** ubicado en la ruta: **app/assets/stylesheets/application.css**

## Generando Vistas

1. Se genera un **layout** según lo vimos anteriormente. En nuestro caso reutilizamos la maqueta de **index.html.erb** y la renombramos: **mashable.index.html**
2. Se configura el route: → **get 'mashable' => 'feeds#mashable'**
3. En el **controlador** se agrega el método:

**class Feed < ActiveRecord::Base**

**def mashable**

**end**

**end**