Laboratorio 2 – Intro RoR, GIT y Heroku

1. Instalar Gestores de Bases de Datos

* Descargar MySQL desde http://dev.mysql.com/downloads/mysql/ (~100MB) y luego instalar
* Descargar SQLite desde http://www.sqlite.org/download.html (~1MB) y luego instalar

1. Instalar Rubygems

* Descargar desde <http://rubygems.org/pages/download>
* Descomprimir e ingresar al directorio creado
* Ejecutar como administrador

**ruby setup.rb**

1. Instalar Rails

* Ejecutar como administrador

**gem install rails**

1. Instalar soporte para Bases de Datos

* Ejecutar como administrador

**gem install mysql**

* Ejecutar como administrador

**gem install sqlite3-ruby**

1. Haciendo un “hola mundo” en RoR

* En una terminal ejecutar

**rails new lab2**

**cd lab2**

**rails g controller lab2 index**

* Editar el archivo lab2/app/views/lab2/index.html.erb con el editor de texto de su preferencia, colocar su nombre en el archivo y luego ejecutar en la terminal

**rails s**

* Acceder por el navegador a <http://localhost:3000/lab2/index>

1. Agregando control de versiones

* Descargar e instalar GIT <http://git-scm.com/>
* Dentro del directorio lab2, ejecutar en una terminal

**git init**

**git add .**

**git commit -m "haciendo el primer commit"**

* Ir a <https://github.com/signup/free> y crear una cuenta gratuita
* Crear un repositorio (<https://github.com/repositories/new>) y llamarle **lab2**
* Ejecutar en una terminal los comandos en **negrillas** (antes de cada uno tiene une breve explicación)

Configurar el nombre y correo, utilizar el mismo correo de registro en github

**git config --global user.name "Juanito Perez"**

**git config --global user.email jperez@gmail.com**

Para poder conectarnos con github necesitamos generar keys de SSH, para hacerlo, siga la explicación para su sistema operativo

<http://help.github.com/msysgit-key-setup/>

<http://help.github.com/linux-key-setup/>

<http://help.github.com/mac-key-setup/>

Luego, es necesario agregar el origen remoto colocando su nombre de usuario y el nombre del repositorio de la forma git@github.com:usuario/repositorio.git esto lo realiza con el comando

**git remote add origin git@github.com:jperez/lab2.git**

Pueden revisar si está agregado con el comando

**git remote show origin**

Hacer “push” hacía este origen remoto con el comando

**git push origin master**

Revisar si fue publicado en el url <http://github.com/jperez/lab2>

reemplazando jperez por su nombre de usuario.

En cualquier momento puede revisarse el estado con el comando

**git status**

y las diferencias con lo que hicieron commit con el comando

**git diff**

Luego, editaremos el archivo lab2/app/views/lab2/index.html.erb con el editor de texto de su preferencia, colocar su carnet debajo de su nombre.

Para agregar los cambios en el control de versiones, pasamos del working directory a staging (con add) y luego con commit se refleja el cambio en el repositorio. Para ello ejecute los siguientes comandos

**git add app/views/lab2/index.html.erb**

**git commit -m "agregando seccion"**

si en caso cometió algún error, es posible revertir el último commit con

**git revert HEAD**

estos cambios puede hacerlos cuantas veces sea necesario, recuerde que GIT

funciona **casi** de forma offline, una vez que esté listo y quiera esos cambios en lo publicado, debe ejecutar

**git push origin master**

1. Publicando en Heroku

* Para “instalar” Heroku es necesaria una gema, ejecute como administrador

**gem install heroku**

* Vaya a <https://api.heroku.com/signup> e inicie una cuenta nueva. Para ver sus aplicaciones puede dirigirse a <https://api.heroku.com/myapps>
* Para crear una aplicación, desde la terminal ejecute **heroku create**

Nota, previo a este paso, debe existir un repositorio de GIT configurado en ese directorio, en esta ocasión ya lo tenemos por el paso anterior y por eso podemos obviarlo.

* Para ver la información de aplicación, desde la terminal ejecute

**heroku info**

* Para publicarlo en heroku, hay que hacer un push pero no al origen, si no a otro lado (en este caso se llama heroku) y lo hace con el siguiente comando

**git push heroku master**

* Para ver su aplicación puede entrar a nombre.heroku.com, en el momento de crear la aplicación le da un nombre aleatorio, vamos a utilizar un estándar l<número de laboratorio>-carnet, por ejemplo l2-20032061 para cambiarlo utilice el comando

**heroku rename l2-20032061l2**