***Instituto Tecnológico de Costa Rica***

***Administración de Tecnologías de Información***

***Curso TI-3404 Lenguajes de Programación***

***Docente***

***Andréi Ricardo Fuentes Leyva***

***Tarea Programada #4 Manual de usuario.***

***Grupo de Trabajo:***

***Estefany Quesada Montero***

***Esteban Aguilar Varverde***

***Jennifer Barrantes***

***Titulo: Aplicación web Ruby­Sinatra***

***Fecha de entrega: Miércoles 19 de junio del 2013***

***Instalación de Ruby (versión 1.8)***

sudo apt­get install ruby1.8

***Instalación de ruby gems***

sudo apt­get install rubygems

***Instalación de conexión con Sinatra***

sudo gem install sinatra

***Instalación de “erb” para uso de HTML como interfaz gráfica web***

De acuerdo al la documentación oficial “ebr” ya esta integrado en la instalación de Ruby. Forma de llamarlo (require 'erb').

***Cómo utilizar “erb” para un ambiente* web.**

Bien, primero que todo, que es “erb”.

“Erb” es una gema de Ruby la cual permite trabajar el lenguaje en sistemas empotrados o en ambientes web esta gema ya esta instalada en la configuración de Ruby por lo que no es necesaria su instalación.

**¿Cómo funciona?**

De forma simple erb permite llamar o visualizar archivos web que poseen la interfaz de usuario, lo realiza mediante una llamada al archivo “erb”.

Ejemplo:

erb :Prueba #Esto llama al archivo Prueba.erbDiferencias entre Ruby y ERB

No existe diferencia pues no son lenguajes diferentes en una forma directa, ya que un programa en Ruby o de formato “rb” contiene código de Ruby en este caso particular contendrá también elementos de codigo ofrecidos por Sinatra mientrea que un archivo formato “erb” contendrá únicamente codigo de HTML, ahora esto no quiere decir que un archivo “erb” no pueda tener código de Ruby ya que si

lo puede tener mediante un agregado a las etiquetas de HTMl.

***Muestra de etiquetas con código Ruby***.

<%\_\_\_\_\_\_\_\_\_\_%> # Código Ruby normal

<%=\_\_\_\_\_\_\_\_\_%> #Muestra de una expresión de Ruby (envía el resultado o valor que contiene la variable, es similar al puts de Ruby pero para TML)

<%#\_\_\_\_\_\_\_\_\_%> # Comentarios de Ruby

El siguiente es un jemplo de código de Ruby dentro de un HTML en un

archivo “erb”. Nota (El fragmento se tomo de forma aleatoria).

<% for j in 0..2 do %>

<% case num

when 2, 8 %>

<td class="colcentro">

<% when 4, 6 %>

<td class="rowcentro">

<% when 5 %>

<td class="colcentro rowcentro">

<% else %>

<td>

<% end %>

***¿Un archivo “erb” se puede tratar como un HTML?***

Por supuesto ya que los “erb” son archivos HTML que están integrados para que puedan usar código de Ruby pero no por eso dejan de ser HTML que pueden ser abiertos por el navegador para revisar que

mostraran al usuario.

Si la información del usuario la recive el “erb”. *¿Cómo la puede manejar Ruby?*

Bien en HTML usted no debe preocuparse tanto de como enviar la información ya que existe un botón solo enviara los datos cada vez que usted lo presione ahora Ruby siempre recibirá esta información como un string (si la información la requiere como otro tipo de dato usted deberá hacer la conversión al tipo deseado)mediante la siguiente notación:

variable\_de\_Ruby = params[:variable\_erb] #Esto por si solo no servirá ya que se requiere una forma de saber con que conector o titulo de Sinatra se esta interactuando por ello

en codigo HTML tenemos:

<form action="/calculos" method="post">

#

#Aquí adentro van los entry y el botón de envío

#

</form>

Bien ahora analizándolo, action contiene el titulo en Sinatra mientras que metod posee una acción esto se vera explicara mejor más adelante, ahora veamos el código en el archivo Ruby para entender la

conexión.

post '/calculos' do

Como en post, aparece en el código comparándolo con Ruby podríamos decir que es como un equivalente a def y '/calculos' como vemos podemos decir que es el nombre del metodo pero este esta entre comillas simples por ser un titulo de sinatra.

***Integración con Sinatra.***

Tal ves este punto ya muchos lo han investigado pero en lo personal esta guía estaría incompleta sin mencionar como funciona lo visto anteriormente en sinatra.

***Instalación de conexión con Sinatra***

sudo gem install sinatra

***Cargar Sinatra en el código de Ruby.***

require 'sinatra'

Ahora el API de Sinatra de forma inmediata empezara la conexión entre nuestro código en Ruby con la pagina de Sinatra despues de conectado el sistema enviara al usuario un puerto local

(generalmente es el puerto 4567).

Sinatra permite realizar varias funciones o trabajos en el código Ruby en esta guia solo veremos tres que creo son las más importantes.

Get – Este permite visualizar la pagina (puede contener o llamar a métodos funcionales).

Post – Este es usado para realizar acciones como llamar métodos o cálculos, no muestra los resultados.

Redirect – Llama a otro titulo de Sinatra en el código de Ruby, los redirect son llamados dentro de otros métodos o un Post.

Ejemplo:

Get:

get '/' do

erb :Prueba

end

Post

post '/calculos' do

$numero1

$numero2

= params[:campo1].to\_i

= params[:campo2].to\_i

end

Redirect:

post '/calculos' do

#

#

#

redirect '/resultado'

end

***¿Cómo sabe Sinatra que mostrar al inicio?***

Es simple, Sinatra siempre buscara el get con el titulo '/' ya que lo tomara como la raíz de todo el programa, y realizara lo que este contenga.

***¿Cómo terminar la conexión?***

Control C y cerrar la pagina del navegador que contenía el programa

***¿Cómo usar el programa?***

***Pantallazos***