Ruby On Rails
para
Programadores Java

Juanjo Bazán
I Conferencia Rails Hispana
Madrid 2006

¿A quién va dirigida?

Buenos Programadores Java

Web

Ya sabes Rails

MVC

Programación Orientada a Objetos

Convenciones sobre configuraciones

MVC

Java.Configuración

Struts

struts-config.xml
web.xml

Spring

interfaces
xml
spring.handler

hibernate

hibernate.cfg.xml
*.hbm.xml

<jsp tags>

Rails ::Convención

Orientación a objetos

- Objetos, mensajes
- Clases
- Interfaces
- Instancias
- Variable/métodos de instancia
- Métodos/variables de clase (static)...
- Herencia

la misma terminología

Mapeo O/R

JDO,
Hibernate,
Toplink...

Active Record

Envoltura sobre filas de BD

Encapsula el acceso a BD

Métodos de clase para obtener instancias

Testing

Usas Junit ?

Con Rails podrás hacer:

Tests unitarios (modelos)

Tests funcionales (controladores)

Test de integración (acciones)

Rails incluye el mejor entorno para testing del 'mercado'. Sin discusión.

¿Ayuda para tareas de desarrollo?

¿Usas Ant?  Usarás Rake sin problemas

Resumen

Todas las convenciones que asume Rails te parecerán naturales.

Rails te proporciona mecanismos simples para las funcionalidades de las herramientas Java que estás acostumbrado a usar.

Rails agrupa funcionalidades que estas acostumbrado a encontrar por separado.

No sabes Ruby

Rails no podría existir sin Ruby

Rails 'hereda' de Ruby su elegancia y simplicidad

Ruby es un lenguaje dinámico orientado a objetos

¿Si llego desde Java qué diferencias me voy a encontrar?

Ruby: Tipado dinámico

Java:

```
int x=2;
x=x^100;
System.out.println(x);   =>  -831846303 (!)
```

Ruby:

```
x=2
x.class  => Fixnum

x=x**100 => 1267650600228229401496703205376
x.class  => Bignum
```

Ruby: ¡No hay tipos básicos!

Java:

```
String.valueOf(33);
Integer.parseInt("33");
```

Ruby:

```
33.to_s      =>  No wrappers!
"33".to_i
11.zero?
7.class => Fixnum
6+2   6.+ 2   6.+(2)
```

Ruby: Todo son objetos

Java:

```
null  => Referencia erronea a un objeto
```

Ruby:

```
x=nil
x.nil?  => true
x.class => NilClass
```

¡¡No NullPointerException!!

Ruby: Manejo simple de Fechas

Java:

```
Date fecha = new
  GregorianCalendar(2006,11,25,16,0).getTime();
```

Ruby:

```
"2006-11-25 16:00".to_time
```

Java:

```
new Date(new Date().getTime() - (30*60*1000))
```

RoR:

```
30.minutes.ago
```

Ruby: Simplicidad de código

Java:

```
public class Conferencia{
   private String nombre;
   private Date fecha;

   public void setNombre(String n){
      this.nombre=n;
   }

   public String getNombre(){
      return nombre;
   }

   public void setFecha(Date f){
      this.fecha=f;
   }

   public Date getFecha(){
      return fecha;
   }
}
```

Ruby:

```
class Conferencia
   attr_accessor :nombre, :fecha
end
```

getter/setter se generan automáticamente

Ruby: Contenedores

Java:

Arrays

Collections Framework:

- ArrayList
- Stack
- List
- HashSet
- HashMap
- ...

Funcionalidades repartidas

(Clases, Interfaces, Metodos estáticos...)

Ruby:

Array	a[1]=x
Hash	h[a]=b
Set	

Todas las funcionalidades es un solo contenedor.

Ruby: Iteradores

Java:

```
for (Enumeration e=parent.getChildren();
  e.hasMoreElements(); )
    { Element child = (Element)e.nextElement();
     // Hacer algo con child
    }
```

Ruby:

```
parent.each_child { |child|
   # Hacer algo con child
}
```

Ruby: Asignación múltiple

Java:

```
int a=2;
int b=3;

int aux;
aux = a;
a = b;
b = aux;
```

Ruby:

```
a=2
b=3

a,b = b,a
```

```
titulo, autor = Libro.getInfo
```

Ruby: es (muy) Dinámico

Puedo añadir métodos a CUALQUIER clase (y en CUALQUIER momento)

```
class String
  def spanish?
    include? “ñ”
  end
end

“Du hast mich”.spanish? => false
```

Ruby: No hay interfaces

No hay herencia de interfaz

¿Y si necesitamos tener métodos comunes?

Módulos

- Agrupación de métodos
- Un módulo no se instancia
- Se incluyen en las clases (include)
- Se convierten en metodos de instancia

OK. ¿Qué consigo a cambio?

Menos código

No XML innecesario

No aprender lenguajes de configuración de apps

Acceso fácil a buenas prácticas: Test, REST,...

Menos sufrimiento

Comunidad efervescente

++ PRODUCTIVIDAD

¡¡ ENSÉÑAMELO !!

Ejemplos de tareas comunes

No hacer nada

Mapear una tabla de BD

Hacer una consulta a BD

Crear un archivo XML

Añadir filtros a una petición web

Hacer una llamada Ajax

Un programa que no hace nada

Java:

Archivo Nada.java:

```
public class Nada{

  public static void main(String[] args){
  }

}
```

Ruby On Rails:

Archivo Nada.rb:

Mapeo O/R

```
create table noticias(
    id int not null autoincrement,
    titular varchar(100),
    entradilla varchar(255),
    texto text,
    primary key(id)
)
```

Java:

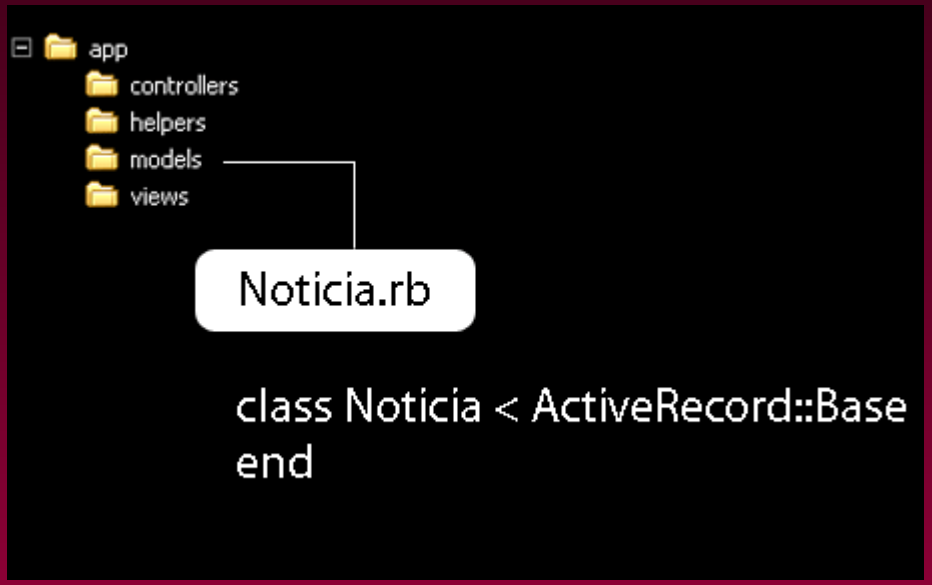
```
public class Noticia{
  private Integer id;
```

```
<?xml version="1.0" ?>
<!DOCTYPE hibernate-mapping (View Source for full doctype...)>
<hibernate-mapping>
  <class name="Noticia" table="noticias">
    <id name="id" type="integer" unsaved-value="null">
      <column name="id" sql-type="int(4)" not-null="true" />
      <generator class="identity" />
    </id>
    <property name="titular" column="titular" type="string" />
```

```
<?xml version='1.0'?>
<!DOCTYPE hibernate-configuration
  SYSTEM "hibernate-configuration-2.0.dtd">

<hibernate-configuration>

  <session-factory>

    <property name="connection.datasource">
                java:comp/env/jdbc/gmsDS</property>
    <property name="show_sql">false</property>
    <property name="dialect">
            net.sf.hibernate.dialect.MySQLDialect</property>

    <!-- Mapping files -->
    <mapping resource="Noticia.hbm.xml"/>
  </session-factory>

</hibernate-configuration>
```

```
}
```

Ruby On Rails:



Consultar la Base de Datos

Java:

```
String titular="Conferencia Rails en Madrid";

Query q=session.createQuery(
     "from noticias n where titular=:t");

q.setParameter(titular);

Noticia noti=(Noticia)q.setMaxResults(1).
                                uniqueResult();
```

Ruby On Rails:

```
noti=Noticia.find(:first, :conditions=>
     [titular="Conferencia Rails en Madrid"] )
```

Crear un archivo XML

Java:

```
import org.dom4j.Document;
import org.dom4j.DocumentHelper;
import org.dom4j.Element;

public class Foo {
   public Document createDocument() {
      Document document = DocumentHelper.
                                          createDocument();
      Element root = document.addElement( "root" );

      Element author1 = root.addElement( "alumno" )
         .addAttribute( "nombre", "Luis" )
         .addAttribute( "apellido", "Crespo" )

      return document;
   }}
```

Ruby On Rails:

```
x = Builder::XmlMarkup.new

x.alumno{
 x.nombre "Luis"
 x.apellido "Crespo"
}
```

```
<alumno>
 <nombre>Luis</nombre>
 <apellido>Crespo</apellido>
</alumno>
```

Método .to_xml !

Añadir un filtro

Java: clase Filter, doFilter, xml

```
public final class MiFlitro implements Filter {
   private FilterConfig filterConfig = null;
   public void init(FilterConfig filterConfig)
      throws ServletException {
      this.filterConfig = filterConfig;
   }
   public void destroy() {
      this.filterConfig = null;
   }
   public void doFilter(ServletRequest request,
      ServletResponse response, FilterChain chain)
          //filtrar algo
    }
```

Ruby On Rails:

```
class MiControladorController < ApplicationController
  before_filter :autorizar

  def autorizar
      #...
  end
end
```

Realizar una llamada Ajax

Java:

```
var req;
function initRequest() {
    if (window.XMLHttpRequest) {
        req = new XMLHttpRequest();
    } else if (window.ActiveXObject) {
        isIE = true;
        req = new ActiveXObject("Microsoft.XMLHTTP");
    }
}

function validateUserId() {
    initRequest();
    req.onreadystatechange = processRequest;
```

```
public  void doGet(HttpServletRequest request, HttpServletResponse  response)
    throws IOException, ServletException {

  String targetId = request.getParameter("id");

  if ((targetId != null) && !accounts.containsKey(targetId.trim())) {
        response.setContentType("text/xml");
        response.setHeader("Cache-Control", "no-cache");
        response.getWriter().write("<valid>true</valid>");
    } else {
        response.setContentType("text/xml");
        response.setHeader("Cache-Control", "no-cache");
        response.getWriter().write("<valid>false</valid>");
    }
  }
```

Ruby On Rails:

```
<html>
  <head>
    <title>Llamada Ajax</title>
    <%= javascript_include_tag "prototype" %>
  </head>
  <body>
    <h1>¿Donde estoy?</h1>
    <div id="div_donde">
      <%= link_to_remote( "Haz click aquí",
          :update => "div_donde",
          :url =>{ :action => :lugar }) %>
      para saberlo.
    </div>
  </body>
</html>
```

```
class EjemploController<ApplicationController
 def lugar
  render_text "<p>En la conferencia Rails!<b>"
 end
end
```

EXTRAS:

Migraciones

RJS

irb

Migraciones

Control de versiones de la BD

No utiliza SQL, utiliza Ruby!

```
create_table "users", :options => 'type=InnoDB' do |t|
  t.column "email", :string, :limit => 100
  t.column "activated_at", :datetime
  t.column "firstname", :string, :limit => 100
  t.column "lastname", :string, :limit => 100
end
```

```
class AddNewEmailToUser < ActiveRecord::Migration
  def self.up
    add_column :users, :new_email, :string, :limit => 100
    add_column :users, :email_change_code, :string
  end

  def self.down
    remove_column :users, :new_email
    remove_column :users, :email_change_code
  end
end
```

RJS

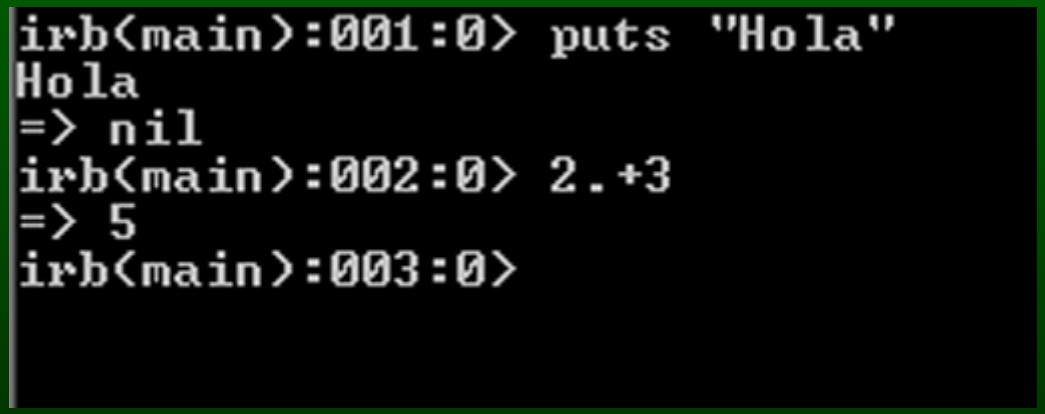
Plantillas Javascript

No se programa en JS sino en Ruby!

```
page.insert_html :bottom, 'list',
                 content_tag("li", "algo")

page.visual_effect :highlight, 'list',
                   :duration => 3

page.replace_html 'header',
                  'RJS Es Fácil!'
```

irb: interactive ruby

Consola de ejecución



Acceso a todas las clases de la aplicación Rails

Resumen

Transición fácil

Aumento de la productividad

Haz feliz al carbono, no al silicio!

¡Gracias!

KFLink

¿dudas? jjbazan@kflink.com