

# Ruby on Rails

## Visión general

Áncor González Sosa

Banot.net  
<http://www.banot.net/>

Febrero de 2011



# Contenidos

- 1 ¿Quienes somos y qué hacemos aquí?
- 2 Antes de entrar en materia
- 3 Ruby
- 4 Ruby on Rails
- 5 En el mundo real



# Contenidos

1 ¿Quienes somos y qué hacemos aquí?

2 Antes de entrar en materia

3 Ruby

4 Ruby on Rails

5 En el mundo real



# ¿Quién soy yo y qué hago aquí?

- ¿Quién es esta gente?
- ¿Qué van a hacernos?
- ¿Por qué?



# ¿Quienes son ustedes?

- ¿Quién es esta gente?
- ¿Qué hace aquí?
- Conocimientos previos
  - ▶ Programación en general
  - ▶ Programación orientada a objetos
  - ▶ Lenguajes dinámicos
  - ▶ Desarrollo web (HTML, CSS, Javascript, AJAX...)
  - ▶ Ingeniería de software (patrones, MVC, desarrollo guiado por tests...)
  - ▶ REST



# Contenidos

1 ¿Quienes somos y qué hacemos aquí?

2 Antes de entrar en materia

3 Ruby

4 Ruby on Rails

5 En el mundo real



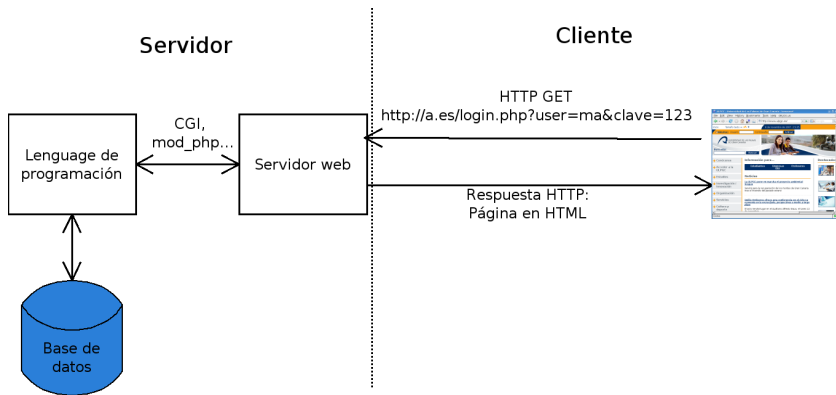
# El manifiesto ágil

Las metodologías ágiles valoran:

- A los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, por encima de la documentación exhaustiva.
- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan.



# Estructura de una aplicación web



# Micro-introducción a la POO

- Clases ~ tipos de variables
- Objetos ~ variables
- Métodos ~ funciones específicas de una clase
- Atributos ~ «variables» dentro de un objeto
- Herencia → especialización de clases
- `foo.bar` → llamar método `bar` del objeto `foo`

**CENSORED**

# Ruby on Rails

Para aplicaciones de alto nivel

***Ruby es el único lenguaje y Rails es su profeta***



# Contenidos

1 ¿Quienes somos y qué hacemos aquí?

2 Antes de entrar en materia

**3 Ruby**

4 Ruby on Rails

5 En el mundo real



# ¿Qué es Ruby?

- Lenguaje de programación muy dinámico
- Tipado *dinámico* y *fuerte*
- De origen *japonés*
- Basado en Perl, Smalltalk y Python, *principalmente*
- Sintaxis con mezcla de ideas de Perl y Python



# ¿Qué es Ruby?

- Lenguaje de programación muy dinámico
- Tipado *dinámico* y *fuerte*
- De origen *japonés*
- Basado en **flexibilidad**, Smalltalk y Python, *principalmente*
- Sintaxis con mezcla de ideas de Perl y Python



# ¿Qué es Ruby?

- Lenguaje de programación muy dinámico
- Tipado *dinámico* y *fuerte*
- De origen *japonés*
- Basado en **flexibilidad**, **orientación a objetos** y Python, *principalmente*
- Sintaxis con mezcla de ideas de Perl y Python



# ¿Qué es Ruby?

- Lenguaje de programación muy dinámico
- Tipado *dinámico* y *fuerte*
- De origen *japonés*
- Basado en **flexibilidad, orientación a objetos y limpieza y alto nivel**, *principalmente*
- Sintaxis con mezcla de ideas de Perl y Python



# Algunos comentarios sobre sintaxis

- Modo «poético» (no hacen falta los paréntesis (bueno, casi nunca (¡en serio!)))
- ¡Mola para hacer DSLs!
- Sufijos de sentencias y `unless`, estilo Perl
- Mayúsculas importantes (constantes)
- Uso de bloques (similar a funciones anónimas de C o *lambda* de Python)



# Pasen y vean

```
class Person
  def initialize(name)
    @name = name
  end
  def say(text)
    puts "<#{@name}> #{text.chomp}"
  end
end
```

```
person = Person.new('Pepito')
person.say('Hola')      # => "<Pepito> Hola"
```



# Ruby es dinámico... mucho

```
hola = "Hola"  
hola.size           # => 4  
hola.cuantas_letras? # => Excepcion undefined method
```

```
class String  
  def cuantas_letras?  
    size  
  end  
end
```

```
hola.cuantas_letras?   # => 4  
"Adios".cuantas_letras? # => 5
```



# Versiones de Ruby

- Lenguaje vs implementación
- Ramas de Ruby (lenguaje)
  - ▶ 1.8
  - ▶ 1.9



# Implementaciones de Ruby

- MRI (*Matz Reference Implementation*)
- Ruby Enterprise Edition (orientado a Rails)
- JRuby (sobre JVM)
- Rubinius (implementado en Ruby)
- MagLev
- MacRuby
- IronRuby (sobre .NET)
- ...



# Contenidos

1 ¿Quienes somos y qué hacemos aquí?

2 Antes de entrar en materia

3 Ruby

**4 Ruby on Rails**

5 En el mundo real



# ¿Qué es Ruby on Rails?

Plataforma para desarrollo de aplicaciones web con Ruby

- Desarrollado por David Heinemeier Hansson, Dinamarca
- Extraído de un proyecto real (Basecamp, de 37signals)
- Facilita el desarrollo, despliegue y mantenimiento
- De bajo nivel pero cercano al dominio del problema
- Basado en prácticas *ágiles* de desarrollo

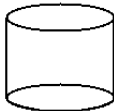


# ¿Por qué es especial?

Don't repeat yourself

```
public class Event {  
  private String title;  
  private Date date;  
  
  public Long getId() {  
    return id;  
  }  
  
  private void setId(Long id) {  
    this.id = id;  
  }  
  
  public Date getDate() {  
    return date;  
  }  
  
  public void setDate(Date date) {  
    this.date = date;  
  }  
}
```

```
CREATE TABLE EVENTS (  
  ...  
  title varchar(256),  
  event_date timestamp,  
  ...  
);
```



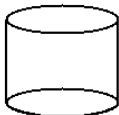
```
<class name="events.Event" table="EVENTS">  
  <id name="id" column="EVENT_ID">  
    <generator class="native"/>  
  </id>  
  <property name="date" type="timestamp" column="EVENT_DATE"/>  
  <property name="title"/>  
</class>
```

# ¿Por qué es especial?

Don't repeat yourself

```
class Event < ActiveRecord::Base
end
```

```
CREATE TABLE EVENTS (
...
  title varchar(256),
  event_date timestamp,
...
);
```



# ¿Por qué es especial?

Convención frente a configuración

```
class Event < ActiveRecord::Base
  set_table_name "event"
  set_primary_key "identifier"
  has_and_belongs_to_many :assistants,
                           :class_name => "People",
                           :join_table => "assist",
                           :foreign_key => "event"
end
```



# ¿Por qué es especial?

Convención frente a configuración

```
class Event < ActiveRecord::Base
  has_and_belongs_to_many :assistants
end
```



# MVC a grandes rasgos

- **M** , **V** y **C**

- Modelo para los datos
- Vista para la interfaz (HTML en este caso)
- Controlador para
- De invención bastante reciente
- Muy popular actualmente para *plataformas* de desarrollo web



# MVC a grandes rasgos

- **M**odelo, **V**ista y **C**ontrolador

- Modelo para los datos
- Vista para la interfaz (HTML en este caso)
- Controlador para
- De invención bastante reciente
- Muy popular actualmente para *plataformas* de desarrollo web



# MVC a grandes rasgos

- **M**odelo, **V**ista y **C**ontrolador
- Modelo para los datos
- Vista para la interfaz (HTML en este caso)
- Controlador para
- De invención bastante reciente
- Muy popular actualmente para *plataformas* de desarrollo web



# MVC a grandes rasgos

- **M**odelo, **V**ista y **C**ontrolador
- Modelo para los datos
- Vista para la interfaz (HTML en este caso)
- Controlador para
  - De invención bastante reciente
  - Muy popular actualmente para *plataformas* de desarrollo web



# MVC a grandes rasgos

- **M**odelo, **V**ista y **C**ontrolador
- Modelo para los datos
- Vista para la interfaz (HTML en este caso)
- Controlador para atraerlos a todos y atarlos a las... errhhh
- De invención bastante reciente
- Muy popular actualmente para *plataformas* de desarrollo web



# MVC a grandes rasgos

- **M**odelo, **V**ista y **C**ontrolador
- Modelo para los datos
- Vista para la interfaz (HTML en este caso)
- Controlador para hacer cálculos y unir todas las piezas
- De invención bastante reciente
- Muy popular actualmente para *plataformas* de desarrollo web



# MVC a grandes rasgos

- **M**odelo, **V**ista y **C**ontrolador
- Modelo para los datos
- Vista para la interfaz (HTML en este caso)
- Controlador para hacer cálculos y unir todas las piezas
- De invención bastante reciente
- Muy popular actualmente para *plataformas* de desarrollo web



# MVC a grandes rasgos

- **M**odelo, **V**ista y **C**ontrolador
- Modelo para los datos
- Vista para la interfaz (HTML en este caso)
- Controlador para hacer cálculos y unir todas las piezas
- De invención bastante reciente (1979)
- Muy popular actualmente para *plataformas* de desarrollo web

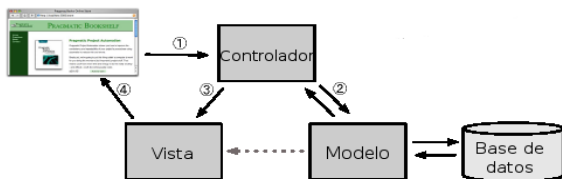


# MVC a grandes rasgos

- **M**odelo, **V**ista y **C**ontrolador
- Modelo para los datos
- Vista para la interfaz (HTML en este caso)
- Controlador para hacer cálculos y unir todas las piezas
- De invención bastante reciente (1979)
- Muy popular actualmente para *plataformas* de desarrollo web



# Arquitectura MVC en una aplicación web



- 1 El navegador envía una petición
- 2 El controlador interactúa con los modelos
- 3 Los controladores invocan a las vistas
- 4 La vista “dibuja” la siguiente página para el navegador



# REST

¿Qué es?

- REST: Representational State Transfer
- Año 2000, Roy Fielding
- Estilo arquitectónico de software para sistemas distribuidos
- Base de la web, tal y como la conocemos...
- ... pero no restringido al protocolo HTTP



# REST

## Principios

- Estado y funcionalidad dividido en **recursos**
- Los recursos cuentan con **interfaces uniformes**:
  - ▶ Identificador global (**nombres**)
  - ▶ Representaciones (tipos de contenido)
- Sobre los recursos se realizan operaciones (**verbos**)
  - ▶ Pocas y bien definidas
  - ▶ De aplicación genérica
  - ▶ Ceñirse a las preexistentes en el protocolo de red
- A través de un protocolo
  - ▶ Cliente/servidor
  - ▶ Sin estado
  - ▶ *Cacheable*
  - ▶ Por capas



# REST

¿Por qué?

- Escalabilidad
- Clientes sencillos
- Simplicidad (menos protocolos implicados)
  - ▶ Más fiable
  - ▶ Comportamiento coherente entre aplicaciones
- Interoperatividad y compatibilidad a largo plazo
  - ▶ HTML para los humanos
  - ▶ JSON y XML para comunicación entre sistemas (webservices *gratis*)
  - ▶ Nuevos clientes añadiendo tipos de contenido
- La web es REST



## ¿Dónde está el IDE?

*"If you're coming to Ruby and Rails from languages such as C# and Java, you may be wondering about IDEs. After all, we all know that it's impossible to code modern applications without at least 100MB of IDE supporting our every keystroke. For you enlightened ones, here's the point in the book where we recommend you sit down, ideally propped up on each side by a pile of framework references and 1,000 page 'Made Easy' books."*

***There are no fully-fledged IDEs for Ruby or Rails.***

*Instead, we all use plain old editors. And it turns out that this isn't as much of a problem as you might think.*

— Dave Thomas



# Contenidos

1 ¿Quienes somos y qué hacemos aquí?

2 Antes de entrar en materia

3 Ruby

4 Ruby on Rails

5 En el mundo real



# Ahí fuera

- Premios y reconocimientos
- Influencia
- Herramientas y bibliotecas
- Ofrecido por muchos proveedores de alojamiento
- Infinidad de aplicaciones y sitios web desarrollados con Rails
  - ▶ Basecamp
  - ▶ Twitter
  - ▶ Yellow Pages
  - ▶ Github
  - ▶ Shopify
  - ▶ Groupon
  - ▶ Scribd
  - ▶ Slideshare
  - ▶ Endeve
  - ▶ BITzen
  - ▶ Jango
  - ▶ Justin TV
  - ▶ New York Jets
  - ▶ A List Apart
  - ▶ Suse Studio
  - ▶ Spree
  - ▶ Redmine
  - ▶ ...



# En NUESTRO mundo real

- Muchas empresas desarrollan con Rails en Gran Canaria
  - ▶ Aentos Consulting
  - ▶ Banot.net
  - ▶ BITzen Solutions
  - ▶ La Caja de Canarias
  - ▶ Fotón Sistemas Inteligentes
  - ▶ Promainsur
  - ▶ Recrea
  - ▶ las que no conocemos...
  - ▶ ...y las que van llegando
- Cursos
- Las Palmas on Rails



# Ruby on Rails

## Visión general

Áncor González Sosa

Banot.net

<http://www.banot.net/>

Febrero de 2011

