

<http://www.trabesoluciones.com>

Soluciones de
internacionalización
con Rails

David Barral Precedo
david.barral@trabesoluciones.com

Asís García Chao
asis.garcia@trabesoluciones.com

Contenidos

- Qué es i18n
 - Por qué es necesaria
- El “no” soporte de Rails
 - Frameworks e i18n
 - Por qué Rails no da soporte
 - Ruby
 - DHH
 - Problemas derivados



Contenidos

- Cómo arreglamos el “marrón”
 - Yo me lo guiso, yo me lo como
 - Lo que guisaron otros: Plugins
 - Nuestra elección
- Conclusiones



Qué es i18n

- Preparar el software para ser usado
 - Con textos “internacionales”
 - En múltiples “locales”
 - Texto traducido (dinámico y estático)
 - Dirección del texto
 - Formato de fechas y números
- Afecta a
 - Diseño
 - Formatos
 - Contenidos (dependientes de culturas)

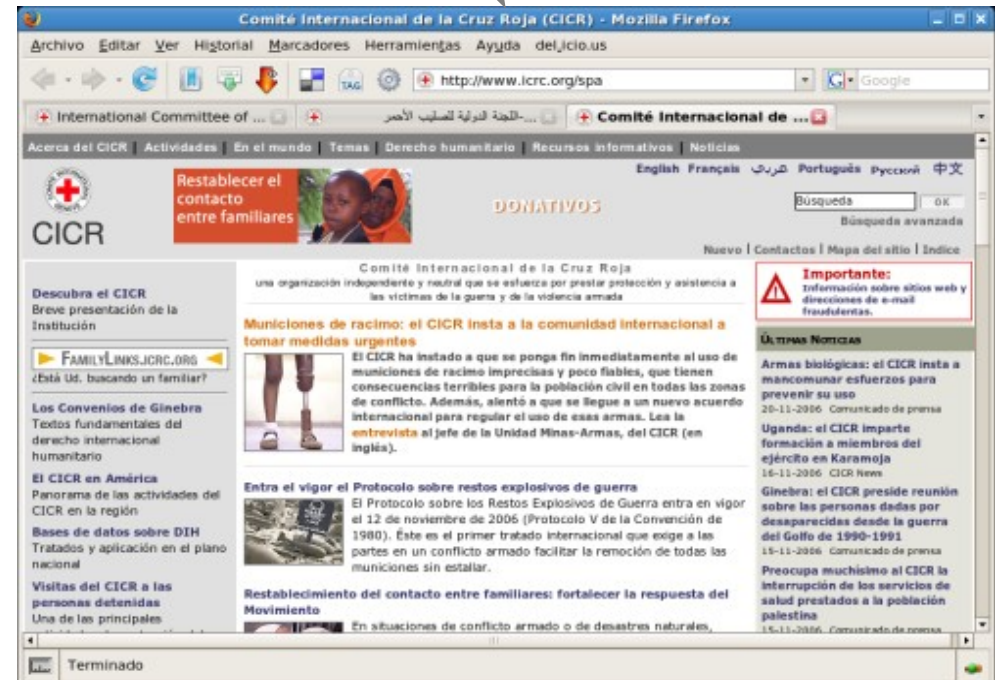
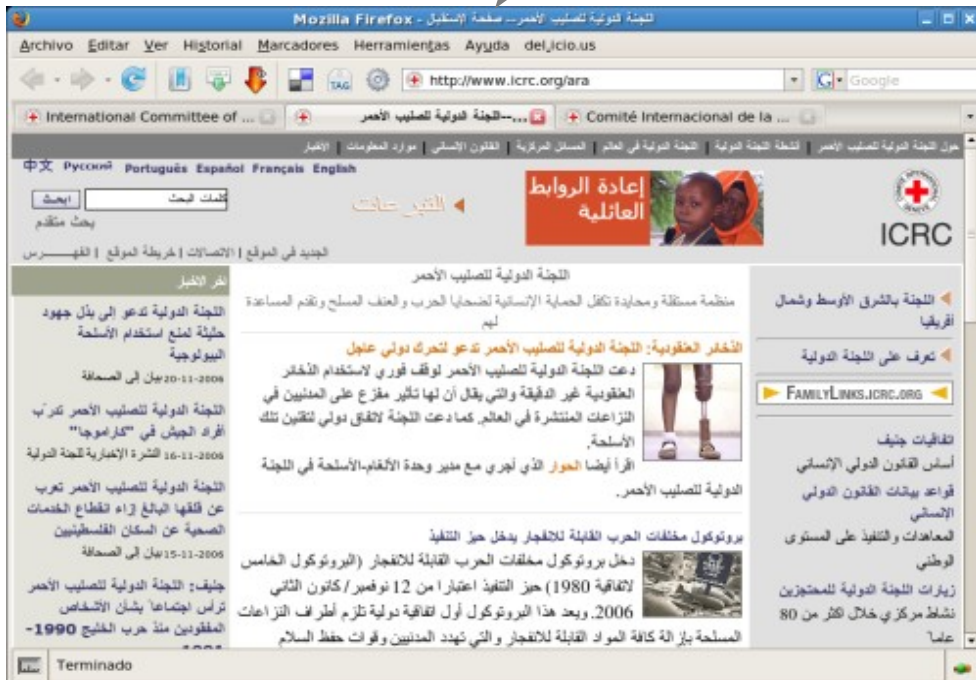


Qué es i18n



Árabe

Castellano



Qué es i18n

- Precio a pagar: mayor complejidad
 - Desarrollo
 - Mantenimiento



Por qué es necesaria la i18n

- Motivos culturales/sociales
 - Acceso a aplicaciones en la propia lengua
 - Acceso universal a internet
 - Difusión de lengua y cultura
- Motivos económicos
 - Llegar a más usuarios



Números gruesos

- Más no-inglés que inglés en el mundo
- Inglés pierde % en internet frente al resto (cada vez más)
- La gente (si puede) usa las aplicaciones en su idioma



Rails e i18n

- Rails es un framework web
- Framework = Software de soporte, desarrollo y explotación de aplicaciones web, que proporciona las características **comunes** a todas ellas
- ¿Es la i18n algo **común** a todas las aplicaciones web?
 - Sí (si tu mercado no es el inglés)



Por qué Rails no da soporte

- Ruby y unicode
 - Librerías y parches
- DHH y cía.
 - No es algo que hayan necesitado (por ahora)
 - No ven claro que la i18n sea **infraestructura** (sino lógica de negocio)



Enfoque DHH

The organized effort is for a Best of Breed™ plugin to emerge and win the hearts of many. There's quite a few contenders out there. Even one in the Rails svn under plugins/localization.

David Heinemeier Hansson



Una opinión inteligente

Incidentally, an engine really wouldn't be a suitable way of doing it - engines are (or should be) focused and relatively self-contained. "Vertical application slices" is a phrase I throw about sometimes. It is far more "horizontal" (i.e. across the whole application like a layer of peanut butter). Engines (and other focussed plugins/components) might want to support it, but not provide it. That's the job of something far more application-wide (or, as it seems to be moving towards, in the core itself).

James Adams (Padre de las Engines)



Problemas derivados

- Las promesas de Rails...
 - Productividad
 - Simplicidad
 - Belleza
 - Felicidad
- ...sólo se cumplen si no necesitas i18n



Problemas derivados

- Múltiples soluciones
 - Adiós a la simplicidad, belleza y felicidad
 - ¿Con cuál me quedo?
 - ¿Y si no he elegido la correcta?
 - ¿Quién resuelve los problemas?
 - Hacking
 - Cómo dar soporte i18n a plugins
 - Elegir una dependencia
 - i18n a mano (más hacking)



Problemas derivados

- El mayor problema de Rails
 - Todo el mundo quiere ser un hacker
 - No todo el mundo puede ser un hacker
- El desarrollo de software debe ser un proceso de ingeniería



```
def cache_add(key, language, idx, translation)
  cache_clear if @cache_size > max_cache_size * 1024
  size = key.size +
    (translation.nil? ? 0 : translation.size)
  @cache_size += size
  @cache[cache_key(key, language, idx)] = translation
end
```



```
def cache_clear
  @cache.clear
  @cache_size = 0
end
```



Cómo arreglamos el “marrón”

- Yo me lo guiso, yo me lo como
 - DRY
 - Necesitas ser un hacker...
 - de Ruby (soporte UTF-8)
 - de Rails
 - y de i18n!!!
 - ¿dónde está la simplicidad? ¿y la belleza?
 - Ya hay muchos hackers...
 - más práctico: colaborar



Cómo arreglamos el “marrón”

- Lo que guisaron otros: Plugins
 - muchas opciones, muchas preguntas



acts_as_translatable

- i18n transparente en objetos del modelo

```
# environment.rb
```

```
TranslationLanguage.base = :es
```

```
class Dog < ActiveRecord::Base
```

```
  acts_as_translatable :fields => [:description]
```

```
end
```



```
Dog.find(:first).description # => "Una perra"
```

```
TranslationLanguage.set_current(:gl)
```

```
Dog.find(:first).description # => "Unha cadela"
```



```
# Nuevos idiomas en el plugin!
```

```
class TranslationLanguage
```

```
  LANGUAGES.merge!({:es => 'español', :gl => 'galego'})
```

```
end
```



```
class TranslationLanguage
  @@current = :en

  #...

  def self.current
    @@current
  end

  def self.set_current(language)
    @@current = self.parse(language)
  end

  #...

end
```



Localize

- Plugin en el repositorio de Rails
- Características
 - Cualquier número de lenguajes y locales
 - Permite definir traducciones para singular/plural
 - Permite definir traducciones “dinámicas” (código Ruby)



```
# /lang/es_ES.rb
```

```
Localization.define('es_ES') do |l|
```

```
  l.store 'yes', 'sí'
```

```
  l.store '%d entries', ['Una entrada', '%d entradas']
```

```
  l.store '(time)', lambda { |t| t.strftime('%H:%M') }
```

```
end
```

```
# /lang/default.rb
```

```
Localization.define do |l|
```

```
  l.store '(time)', lambda { |t| t.strftime('%I:%M') }
```

```
end
```



```
# Controlador o application.rb
```

```
Localization.lang = 'es_ES'
```

```
# Vista
```

```
<%=_ 'yes' %> # => 'sí'
```

```
<%=_ '%d entries', 1 %> # singular => 'Una  
entrada'
```

```
<%=_ '%d entries', 4 %> # plural => '4 entradas'
```

```
<%=_ '(time)', Time.now %> # llama al bloque
```



Localize

- ¿Donde está....
 - la belleza?
 - la simplicidad?
 - la felicidad?



GLoc

- Traducción / localización
 - un fichero .yml por idioma
 - clave: valor
 - reglas
 - helpers
- Rails
 - helpers
 - validación
 - vistas ActionMailer



RAILS_ROOT/lang/es.yml

Aplicación

product.name: nombre

product.description: descripción

#...

Rails

actionview_datehelper_select_month_names:

Enero, Febrero, Marzo, Abril, Mayo, Junio, Julio,
Agosto, Septiembre, Octubre, Noviembre, Diciembre

actionview_datehelper_select_month_names_abbr:

Ene, Feb, Mar, Abr, May, Jun, Jul, Ago, Sep, Oct, Nov, Dec

#...



```
# environment.rb
GLoc.set_config :default_language => :es
GLoc.set_config :raise_string_not_found_errors => false
GLoc.set_config :default_param_name => 'locale'
# UTF-8
GLoc.set_kcode
GLoc.load_localized_strings

class Object
  def _(key, *arguments)
    self.l(key, *arguments)
  end
end
```



```
#routes.rb
```

```
map.connect ':locale/:controller/:action/:id'
```

```
#application.rb
```

```
class ApplicationController < ActionController::Base  
  autodetect_language_filter :check_accept_header => true
```



```
<label for="product_name"><%= _('product_name' %)></label>
```

```
# modelo
```

```
class Product < ActiveRecord::Base  
  validates_size_of :name, :minimum => 5,  
    :message => :product_name_too_short  
end
```

```
# RAILS_ROOT/lang/es.yml
```

```
# Aplicación
```

```
product_name: nombre
```

```
product_description: descripción
```

```
product_name_too_short: nombre tiene menos de %d letras
```

```
#...
```



```
# custom rule
```

```
_gloc_rule_custom: ' |n| return "_a_lot" if n > 5000;  
  "_less"; '
```

```
dogs_count_less: %d perros
```

```
dogs_count_a_lot: ¡¡¡Una jauría impresionante!!!
```

```
people_count_less: %d personas
```

```
people_count_a_lot: Una muchedumbre
```

```
lwr_( :custom, :dogs_count, 5) # => 5 perros
```

```
lwr_( :custom, :people_count, 10000) # => una muchedumbre
```



Globalize

- Fácil de integrar en nuestra aplicación
- Traducción
 - Vistas y modelos (contenido BBDD)
 - Plurales
 - Selección de plantillas por locale
 - ActionView
 - ActionMailer
- Formatos numéricos, de fecha, de moneda, etc.
- +7500 idiomas y 239 países



```
$ .script/plugin install globalize  
$ rake globalize:setup
```



Globalize

- Tabla `globalize_countries`
 - Contiene información sobre los países
 - Código ISO 3166 (2 letras) del país
 - Nombre (en inglés)
 - Formatos (fecha, moneda, numéricos)



Globalize

- Tabla globalize_languages
 - Información sobre los idiomas
 - Códigos ISO (2 y 3 letras)
 - Nombre (inglés)
 - Nombre (en el propio idioma)
 - Dirección de escritura
 - Fórmula para el cálculo de plurales



Globalize

- Tabla globalize_translations
 - Información traducida
 - ModelTranslation: modelos
 - ViewTranslation: mensajes



```
# environment.rb
```

```
include Globalize
```

```
Locale.set_base_language 'en-US'
```

```
LOCALES = { 'en' => 'en-US',  
            'gl' => 'gl-ES' }.freeze
```

```
# routes.rb
```

```
map.connect ':locale/:controller/:action/:id'
```



```
before_filter :set_locale
  def set_locale
    if !params[:locale].nil? &&
      LOCALES.keys.include?(params[:locale])
      Locale.set LOCALES[params[:locale]]
    else
      redirect_to params.merge(
        'locale' => Locale.base_language.code )
      return false
    end
  end
end
```



```
<%= link_to 'es', params.merge!({  
    :controller => controller.controller_name,  
    :action => controller.action_name,  
    :locale => 'es'}) %>
```



```
class Dog < ActiveRecord::Base
  translates :description
end
```

```
Locale.set('en-US')
```

```
Dog.create(:name => 'Fuco', :description => 'a
dog')
```

```
Locale.set('gl-ES')
```

```
d = Dog.find(1) # 'Fuco', 'a dog'
```

```
d.description = 'un can'
```

```
d.save # en-US => 'a dog', gl-ES => 'un can'
```



```
<%= "Add your dog".t %>
```

```
<!-- gl_ES => 'Engade o teu can' -->
```

```
<%= 23459.60.localize %>
```

```
<!-- gl_ES => 23.459,60 -->
```

```
<%= dog.birth_date.to_date.localize('%d/%m/%Y')  
%>
```

```
<!-- gl_ES => 19/04/1979 -->  
country = Country.find(:first, :conditions => "code='ES'")  
country.update_attribute(:date_format, '%d/%m/%Y')
```

```
<%= soldier.birth_date.to_date.localize(  
  Locale.country.date_format) %>
```



Globalize

- Errores Active Record
 - Traducir combinación (campo, mensaje error)
- Moneda
 - Tipo Currency
 - Formato sin usar en BD
 - Soporte completo en BD (Buggy)



Localization Simplified

- Aplicación en un idioma distinto del inglés
- Soporta varios idiomas (alemán, español, francés, italiano, danés, holandés, sueco, etc.)



Localization Simplified

- HTTP content-type => UTF-8
- Conexión a la BBDD => UTF-8
- Rails
 - date_select, etc. (nombres y orden)
 - Errores ActiveRecord
 - distance_of_time_in_words
 - to_currency
 - plurales



Localization Simplified

- Usa métodos Rails estándar => ok
- Sencillo de instalar

```
# plugin init.rb  
require 'lang_es'
```



Localization Simplified

Nuevo perro

2 errores no permite guardar dog

Ha habido problemas con los siguientes campos:

- Name no puede estar en blanco
- Description no puede estar en blanco

Nombre

Descripción



GetText

- Opción más usada (“universalmente”)
 - Herramientas para traducción
 - Soporte Rails
 - Etc.
- Más información... en la siguiente charla



Nuestra elección

- Globalize o Gettext, dependiendo del caso
- Localization Simplified, para aplicaciones en un **único** idioma distinto del inglés



Conclusiones

- Muchas opciones para internacionalizar...
...cada opción enfocada a un tipo de necesidad,
pero solapadas...
...pero pocas funcionan completamente (al menos
no al nivel que se requiere para una aplicación
compleja)
- Internacionalización en el core: estándar



Conclusiones

that i18n is:

- 1. Actually important to many people*
- 2. Not really the difficult to implement in concept*
- 3. Something that people like to point out about Rails to say it's not ready for "Enterprise"*



Referencias

- Wiki Rails, comparativa i18n

<http://wiki.rubyonrails.org/rails/pages/InternationalizationComparison>

- act_as_translatable

<http://blog.wireless.org.ua/articles/2006/10/30/acts-as-translatable-ruby-on-rails-plugin>

- Foro Rails i18n

<http://www.ruby-forum.com/forum/20>

