

De por qué ED debió migrar a la web

Ventajas de usar Frameworks y herramientas web para distribuir soluciones de desarrollo

Carlos Ledezma

Akdemia

15/05/2014

1 Acerca de mí

2 La historia de ED

- Presentación de ED
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- La conclusión evidente

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- Algunos extras
- La conclusión evidente

4 Despedida y preguntas

1 Acerca de mí

2 La historia de ED

- Presentación de ED
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- La conclusión evidente

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- Algunos extras
- La conclusión evidente

4 Despedida y preguntas

Acerca de mí

- ¿Por qué les hablo acerca de mí?

Acerca de mí

- ¿Por qué les hablo acerca de mí?
- ¿Dónde trabajo?

akademia

Acerca de mí

- ¿Por qué les hablo acerca de mí?
- ¿Dónde trabajo?

akademia

- ¿De dónde vengo?



1 Acerca de mí

2 La historia de ED

- Presentación de ED
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- La conclusión evidente

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- Algunos extras
- La conclusión evidente

4 Despedida y preguntas

1 Acerca de mí

2 La historia de ED

- **Presentación de ED**
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- La conclusión evidente

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- Algunos extras
- La conclusión evidente

4 Despedida y preguntas

- Recién graduado.
- Inteligente.
- Emprendedor.
- Le gustan los retos.
- Poca experiencia laboral.
- Sólo código universitario.
- Muchas dudas.



ED tiene una idea



- Encontró un nicho no explotado.
- Tiene un cliente potencial, C.
- Le desarrollará P a C.

ED tiene una idea



- Encontró un nicho no explotado.
- Tiene un cliente potencial, C.
- Le desarrollará P a C.

EMPIEZA EL DESARROLLO!!!

1 Acerca de mí

2 La historia de ED

- Presentación de ED
- **El proceso de desarrollo**
- La entrega de la solución
- La etapa de soporte
- La conclusión evidente

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- Algunos extras
- La conclusión evidente

4 Despedida y preguntas

Acerca de mí
La historia de ED
Ayudando a ED
Despedida y preguntas

Presentación de ED
El proceso de desarrollo
La entrega de la solución
La etapa de soporte
La conclusión evidente

Escoger un lenguaje



Acerca de mí
La historia de ED
Ayudando a ED
Despedida y preguntas

Presentación de ED
El proceso de desarrollo
La entrega de la solución
La etapa de soporte
La conclusión evidente

Escoger un lenguaje

C++
#include <iostream>
using namespace std;

int main() {
 cout << "Hola Facebook!!";
 return 0;
}



Acerca de mí
La historia de ED
Ayudando a ED
Despedida y preguntas

Presentación de ED
El proceso de desarrollo
La entrega de la solución
La etapa de soporte
La conclusión evidente

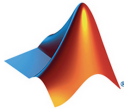
Escoger un lenguaje



Acerca de mí
La historia de ED
Ayudando a ED
Despedida y preguntas

Presentación de ED
El proceso de desarrollo
La entrega de la solución
La etapa de soporte
La conclusión evidente

Escoger un lenguaje



Comunicación con la base de datos

- sqlapi

Comunicación con la base de datos

- sqlapi → GCC.

Comunicación con la base de datos

- sqlapi → GCC.
- MySQL connector

Comunicación con la base de datos

- sqlapi → GCC.
- MySQL connector → ED quiere PostgreSQL.

Comunicación con la base de datos

- sqlapi → GCC.
- MySQL connector → ED quiere PostgreSQL.
- libodbc

Comunicación con la base de datos

- sqlapi → GCC.
- MySQL connector → ED quiere PostgreSQL.
- libodbc → Aún en desarrollo.

Comunicación con la base de datos

- sqlapi → GCC.
- MySQL connector → ED quiere PostgreSQL.
- libodbc → Aún en desarrollo.
- libpqxx.

Interfaz gráfica

- Qt.
- gtkmm.
- wxWidgets.
- FLTK.

Diseño estructural

Implementar diseño en capas:

- Modelo → Encapsular llamadas a DB.
- Controlador → Crear puentes de comunicación.
- Vista → Comunicar con librería gráfica.
- Debe haber total separación.



1 Acerca de mí

2 La historia de ED

- Presentación de ED
- El proceso de desarrollo
- **La entrega de la solución**
- La etapa de soporte
- La conclusión evidente

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- Algunos extras
- La conclusión evidente

4 Despedida y preguntas

Colocando todo en el pendrive

ED compila su producto:

- ¿Arquitectura correcta?
- Hacer instalador.
- ¿Requerimientos en el cliente?
 - Manejador de base de datos.
 - Librerías gráficas.
 - ¿Documentados requerimientos?
 - Lidar con máquinas diferentes.

1 Acerca de mí

2 La historia de ED

- Presentación de ED
- El proceso de desarrollo
- La entrega de la solución
- **La etapa de soporte**
- La conclusión evidente

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- Algunos extras
- La conclusión evidente

4 Despedida y preguntas

Los primeros errores



- Primer intento → teléfono.

Los primeros errores



- Primer intento → teléfono.
- JA

Los primeros errores



- Primer intento → teléfono.
- **JA**
- ED debe buscar sus bitácoras.

Hora de analizar



Hora de analizar



Posibles errores:

- Librería de comunicación.

Hora de analizar



Posibles errores:

- Librería de comunicación.
- Manejo de memoria.

Hora de analizar



Posibles errores:

- Librería de comunicación.
- Manejo de memoria.
- Particulares de la arquitectura.

A solucionar los problemas

- Pasar un par de meses corrigiendo.
- Recompilar.
- Crear instalador.
- Llevaron al cliente.



A solucionar los problemas

- Pasar un par de meses corrigiendo.
- Recompilar.
- Crear instalador.
- Llevaron al cliente.



Proceso se repite con cada error

A solucionar los problemas

- Pasar un par de meses corrigiendo.
- Recompilar.
- Crear instalador.
- Llevaron al cliente.



Proceso se repite con cada error → Soluciones lentas

1 Acerca de mí

2 La historia de ED

- Presentación de ED
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- **La conclusión evidente**

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- Algunos extras
- La conclusión evidente

4 Despedida y preguntas

ED pierde a su cliente



1 Acerca de mí

2 La historia de ED

- Presentación de ED
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- La conclusión evidente

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- Algunos extras
- La conclusión evidente

4 Despedida y preguntas

1 Acerca de mí

2 La historia de ED

- Presentación de ED
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- La conclusión evidente

3 Ayudando a ED

- **Sugerencia y presentación de la herramienta**
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- Algunos extras
- La conclusión evidente

4 Despedida y preguntas

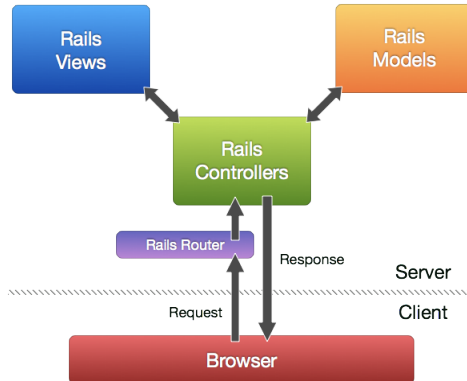
Salvemos a ED usando RoR

¿Qué es Ruby on Rails?

- Framework de desarrollo web.
- Basado en Ruby.
- M.V.C.
- CoC
- Incluye todos los procesos del desarrollo web.
- Apoyado por gemas.



El modelo M.V.C. de RoR



Convenciones sobre la configuración

Lo básico:

Convenciones sobre la configuración

Lo básico:
app/models/:

- ActiveRecord::Base.
- Clases ↔ Tablas.
- Infinidad de métodos predefinidos.

Convenciones sobre la configuración

Lo básico:

app/models/:

- ActiveRecord::Base.
- Clases ↔ Tablas.
- Infinidad de métodos predefinidos.

```
# app/models/user.rb
class User < ActiveRecord::Base
  belongs_to :company # modelo Company
  has_many :associates # modelo Associate
  has_one :session # modelo Sessions
  ...
end
```

Convenciones sobre la configuración

Lo básico:

app/models/:

- ActiveRecord::Base.
- Clases ↔ Tablas.
- Infinidad de métodos predefinidos.

app/controllers/:

- ActionController::Base.
- Manejan la lógica.
- Reciben peticiones.

```
# app/models/user.rb
class User < ActiveRecord::Base
  belongs_to :company # modelo Company
  has_many :associates # modelo Associate
  has_one :session # modelo Sessions

  ...
end
```


Convenciones sobre la configuración

Lo básico:

app/models/:

- ActiveRecord::Base.
- Clases ↔ Tablas.
- Infinidad de métodos predefinidos.

app/controllers/:

- ActionController::Base.
- Manejan la lógica.
- Reciben peticiones.

```
# app/models/user.rb
class User < ActiveRecord::Base
  belongs_to :company # modelo Company
  has_many :associates # modelo Associate
  has_one :session # modelo Sessions
end

...

# app/controllers/users_controller.rb
class UsersController < ActionController::Base
  def index
    @users = User.where( ... )
  end

  ...
end
```

Convenciones sobre la configuración

Lo básico:

app/models/:

- ActiveRecord::Base.
- Clases ↔ Tablas.
- Infinidad de métodos predefinidos.

app/controllers/:

- ActionController::Base.
- Manejan la lógica.
- Reciben peticiones.

app/views:

- Archivos .erb.
- Contienen las interfaces.
- HTML, Haml.

```
# app/models/user.rb
class User < ActiveRecord::Base
  belongs_to :company # modelo Company
  has_many :associates # modelo Associate
  has_one :session # modelo Sessions
end

...

# app/controllers/users_controller.rb
class UsersController < ActionController::Base
  def index
    @users = User.where( ... )
  end

  ...
end
```

Convenciones sobre la configuración

Lo básico:

app/models/:

- ActiveRecord::Base.
- Clases ↔ Tablas.
- Infinidad de métodos predefinidos.

app/controllers/:

- ActionController::Base.
- Manejan la lógica.
- Reciben peticiones.

app/views:

- Archivos .erb.
- Contienen las interfaces.
- HTML, Haml.

```
# app/models/user.rb
class User < ActiveRecord::Base
  belongs_to :company # modelo Company
  has_many :associates # modelo Associate
  has_one :session # modelo Sessions
end

...

# app/controllers/users_controller.rb
class UsersController < ActionController::Base
  def index
    @users = User.where( ... )
  end

  ...
end

<!-- app/views/users/index.html.erb -->
...
<%= @users.each do |user| %>
  <p> <%= user.name %> </p>
<% end %>
...

```

Apoyo de las gemas (demos)



Apoyo de las gemas (demos)



- Squeel → Mejor SQL.

Apoyo de las gemas (demos)



- Squeel → Mejor SQL.
- Pry → Depuración.

Apoyo de las gemas (demos)



- Squeel → Mejor SQL.
- Pry → Depuración.
- CanCan → Autorización.

Apoyo de las gemas (demos)



- Squeel → Mejor SQL.
- Pry → Depuración.
- CanCan → Autorización.
- Authlogic → Autenticación.

Apoyo de las gemas (demos)



- Squeel → Mejor SQL.
- Pry → Depuración.
- CanCan → Autorización.
- Authlogic → Autenticación.
- simple_form, prawn, nested_form, spreadsheet, haml, zeus, jquery-rails...

INVIERTE TU TIEMPO EN TU PROYECTO

Acerca de mí
La historia de ED
Ayudando a ED
Despedida y preguntas

Sugerencia y presentación de la herramienta
El proceso de desarrollo
La entrega de la solución
La etapa de soporte
Algunos extras
La conclusión evidente

Ahora, volvamos con ED



1 Acerca de mí

2 La historia de ED

- Presentación de ED
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- La conclusión evidente

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- **El proceso de desarrollo**
- La entrega de la solución
- La etapa de soporte
- Algunos extras
- La conclusión evidente

4 Despedida y preguntas

Todo es tan fácil en RoR



Todo es tan fácil en RoR



- Lenguaje → Ruby + RoR.

Todo es tan fácil en RoR



- Lenguaje → Ruby + RoR.
- Comunicación con B.D. → pg + database.yml.

Todo es tan fácil en RoR



- Lenguaje → Ruby + RoR.
- Comunicación con B.D. → pg + database.yml.
- Interfaz → HTML + CSS.

Todo es tan fácil en RoR



- Lenguaje → Ruby + RoR.
- Comunicación con B.D. → pg + database.yml.
- Interfaz → HTML + CSS.
- Estructura → MVC.

1 Acerca de mí

2 La historia de ED

- Presentación de ED
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- La conclusión evidente

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- El proceso de desarrollo
- **La entrega de la solución**
- La etapa de soporte
- Algunos extras
- La conclusión evidente

4 Despedida y preguntas

¿En un pendrive?



¿En un pendrive?



- ¿Compilación? → Ruby interpretado.

¿En un pendrive?



- ¿Compilación? → Ruby interpretado.
- ¿Instalador? → Entregar vía web:
 - Heroku.
 - Amazon Web Services.
 - ...

¿En un pendrive?



- ¿Compilación? → Ruby interpretado.
- ¿Instalador? → Entregar vía web:
 - Heroku.
 - Amazon Web Services.
 - ...
- Requerimientos del cliente → Un navegador + internet.

1 Acerca de mí

2 La historia de ED

- Presentación de ED
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- La conclusión evidente

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- El proceso de desarrollo
- La entrega de la solución
- **La etapa de soporte**
- Algunos extras
- La conclusión evidente

4 Despedida y preguntas

Los errores van a llegar



Primer intento

Análisis

Los errores van a llegar



Primer intento

- Teléfono.

Análisis

Los errores van a llegar



Primer intento

Análisis

- Teléfono. **JAJA.**

Los errores van a llegar



Primer intento

- Teléfono. **JAJA.**
- Soporte remoto, en vivo.
- No hay que buscar en el cliente → Información en servidor.

Análisis

Los errores van a llegar



Primer intento

- Teléfono. **JAJA.**
- Soporte remoto, en vivo.
- No hay que buscar en el cliente → Información en servidor.

Análisis

- Mayoría de errores en lógica implementada.

Los errores van a llegar



Primer intento

- Teléfono. **JAJA.**
- Soporte remoto, en vivo.
- No hay que buscar en el cliente → Información en servidor.

Análisis

- Mayoría de errores en lógica implementada.
- Poco probables:
 - Librerías.
 - Memoria.
 - Arquitectura.

Soluciones al instante



- Par de meses corrigiendo.
- Lanzamientos cada semana.
- **Sin** compilar.
- **Sin** instalador.
- Remotamente.

Soluciones al instante



- Par de meses corrigiendo.
- Lanzamientos cada semana.
- **Sin** compilar.
- **Sin** instalador.
- Remotamente.

Soluciones rápidas

1 Acerca de mí

2 La historia de ED

- Presentación de ED
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- La conclusión evidente

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- **Algunos extras**
- La conclusión evidente

4 Despedida y preguntas

El valor agregado

¡La aplicación está en la nube!

- Crear APIs.
- Agregar servicios (SaaS).
- Seguridad física.
- Ampliación de funcionalidad.
- Accesibilidad.



1 Acerca de mí

2 La historia de ED

- Presentación de ED
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- La conclusión evidente

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- Algunos extras
- **La conclusión evidente**

4 Despedida y preguntas

Éxito!



1 Acerca de mí

2 La historia de ED

- Presentación de ED
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- La conclusión evidente

3 Ayudando a ED

- Sugerencia y presentación de la herramienta
- El proceso de desarrollo
- La entrega de la solución
- La etapa de soporte
- Algunos extras
- La conclusión evidente

4 Despedida y preguntas

Conclusiones

CONCLUSIONES

Preguntas

