



# Clase 16:

## Final



profesor: **Patricio López Juri** { [patricio@lopezjuri.com](mailto:patricio@lopezjuri.com) }  
créditos: **10**  
horario: **J:7-8**  
sala: **H3**



**Supuesto que hago  
yo en este curso:**

*Ustedes son personas inteligentes y pueden aprender y deducir reglas lógicas en base a ejemplos.*



# Ejercicio

English:

- How much is that red umbrella?
- How much is that small camera?

Japanese:

- Ano akai kasa wa ikura desu ka.
- Ano chiisai kamera wa ikura desu ka.



# Ejercicio

English:

- How much is that **red umbrella?**
- How much is that **small camera?**

Japanese:

- Ano akai kasa wa ikura desu ka.
- Ano chiisai kamera wa ikura desu ka.



# Ejercicio

English:

- How much is that **red umbrella**?
- How much is that **small camera**?

Japanese:

- Ano **akai kasa** wa ikura desu ka.
- Ano **chiisai kamera** wa ikura desu ka.

Que descubrimos y aprendimos?

- red umbrella = akai kasa = paraguas rojo
- small camera = chiisai kamera = small camera

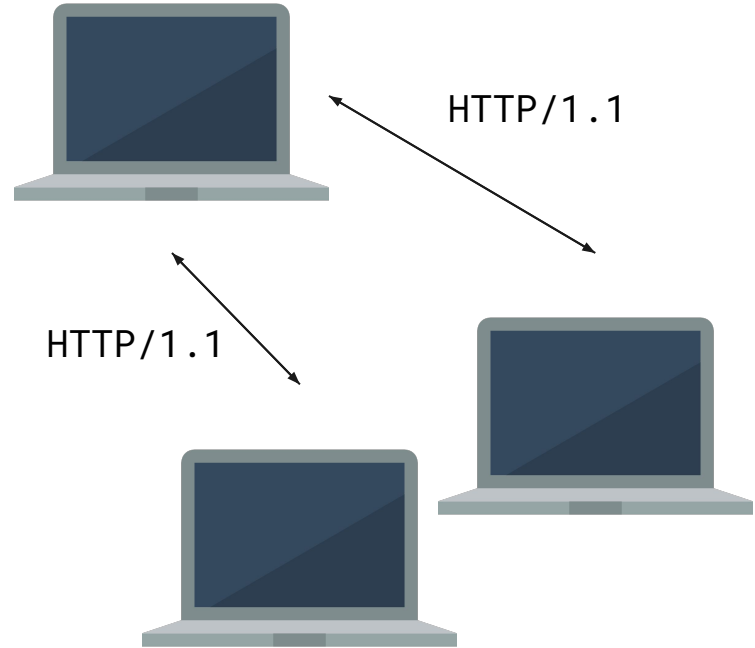
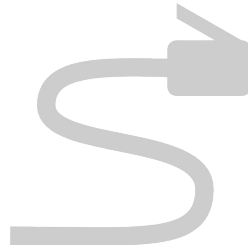


# Qué hemos aprendido del curso

- La diferencia entre la Web y el Internet.
- Que la Web funciona sobre HTTP y usa métodos como GET, POST, PUT, DELETE, etc.
- Sitios Web estáticos con HTML y CSS.
- Frameworks CSS como Bootstrap.
- La importancia del SEO y cómo mejorarlo.
- Algo de Git, o que al menos existe.
- **Aplicaciones Web dinámicas con Ruby on Rails**
- Que existen IaaS, PaaS y en particular la importancia de los SaaS.
- Existen disciplinas y metodologías para llevar a cabo un proyecto TI.
- Conceptos básicos de UX (user experience) y que no es lo mismo que la UI.
- Se puede programar lógica como juegos o visualizaciones en los browsers usando Javascript.

# Web - Internet

No es lo mismo, la web está construída sobre el Internet y usa HTTP. El internet es solo computadores conectados.





## MOBILE FIRST

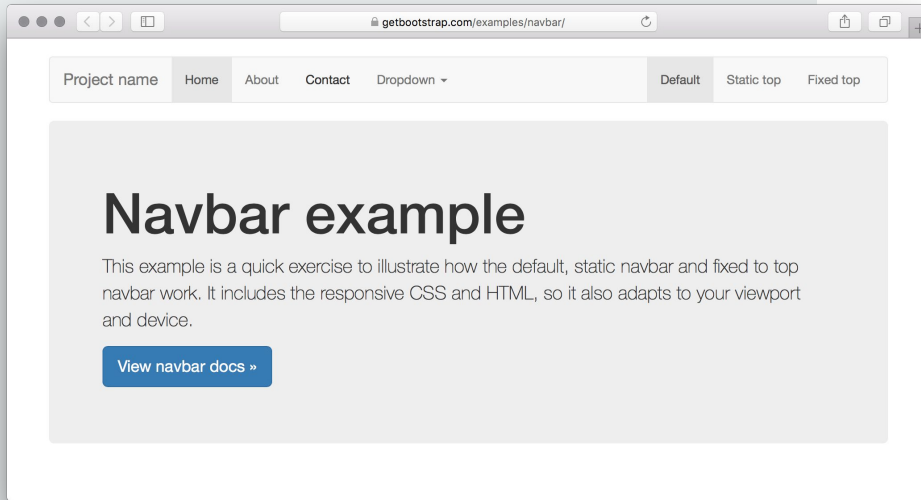
La mayoría de la gente accede a los sitios web **desde los celulares**.

Para el **2018** se estima que más del **80%** vendrá desde celulares.





# Bootstrap



<https://getbootstrap.com/>

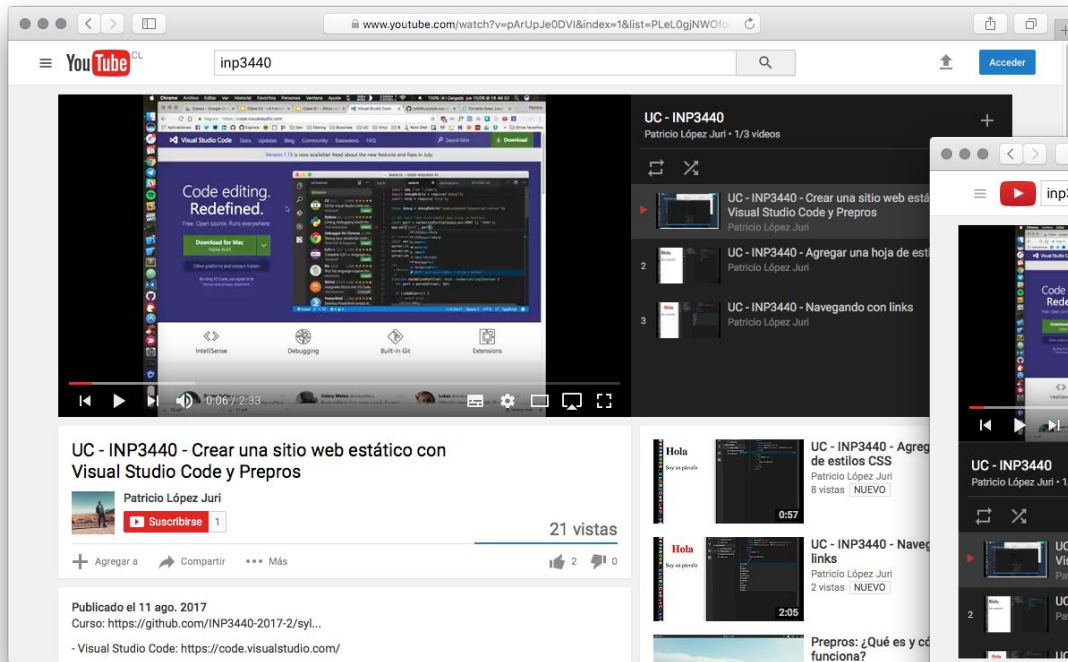




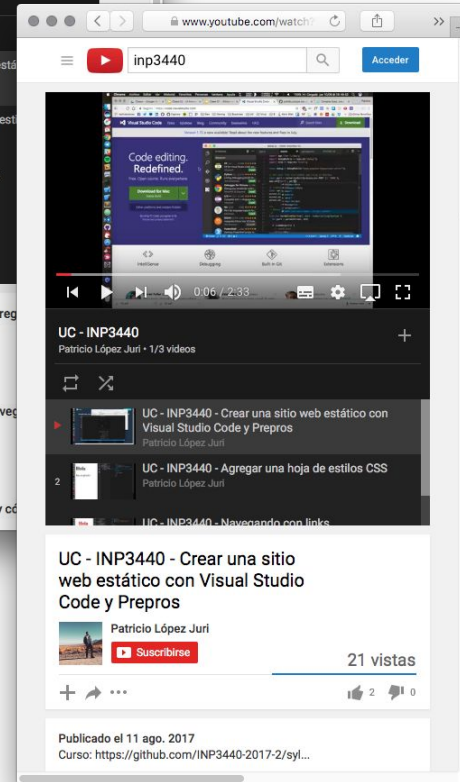
## Mobile First: *Responsive* vs. *Adaptative*

¿Podemos tener una página web *adaptative* y *responsive* al mismo tiempo?

## Escritorio grande



## Escritorio pequeño

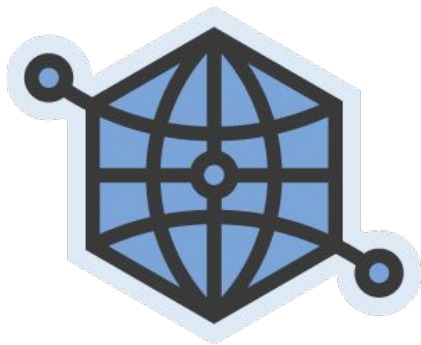


## Celular pequeño

Claro, el ejemplo de YouTube!

# The OpenGraph protocol

- Facebook desarrolló este protocolo.
- Permite a los sitios web mejor visibilidad y mayor atractivo.
- <http://ogp.me/>



<https://blog.ghost.org/content/images/2014/11/structurepost.gif>

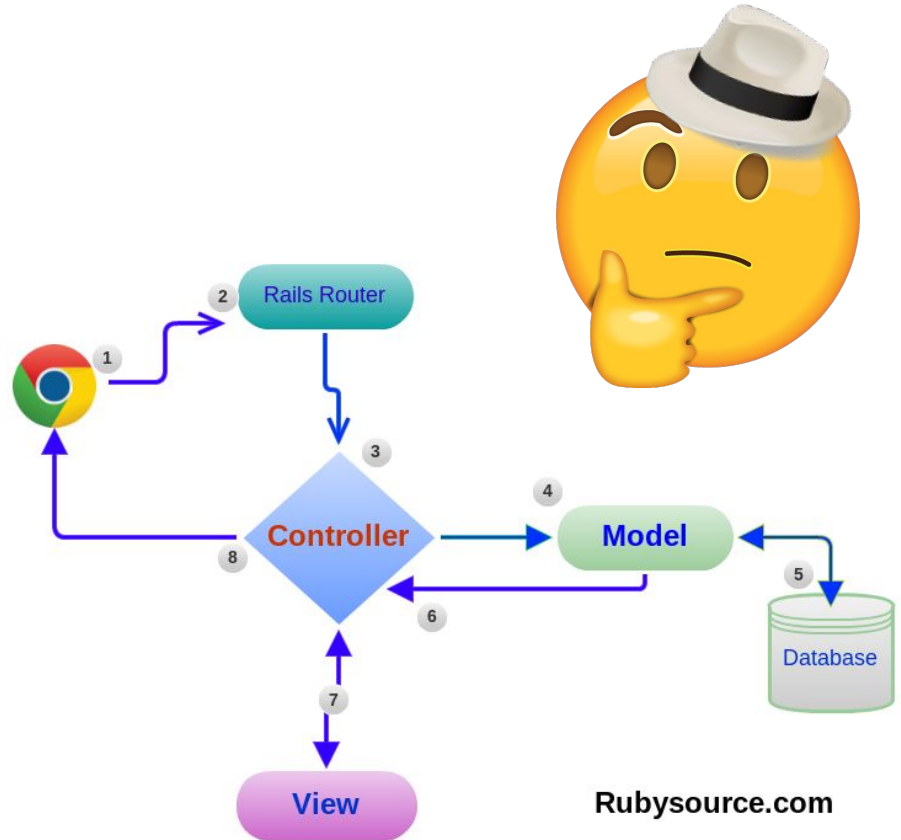


—

# Ruby on Rails

En orden de ejecución:

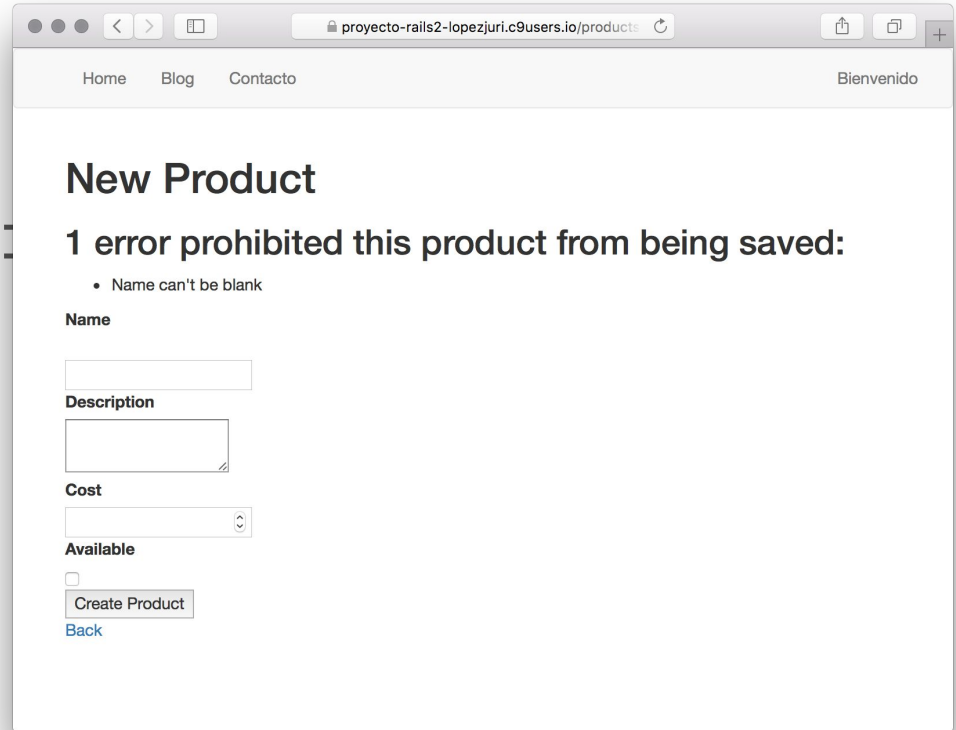
1. *Router*
2. *Controller* (controlador)
3. *Model* (modelo)
4. *View* (vista)



Rubysource.com

# Validadores

```
class Product < ActiveRecord::Base
  validates :name,
end
```



The screenshot shows a web browser window with the address bar displaying 'proyecto-rails2-lopezjuri.c9users.io/products'. The page has a navigation bar with links for 'Home', 'Blog', and 'Contacto', and a 'Bienvenido' message on the right. The main content area is titled 'New Product' and displays a validation error: '1 error prohibited this product from being saved:'. Below this, a bulleted list indicates the error: '• Name can't be blank'. The form contains several input fields: 'Name' (which is empty), 'Description' (a text area), 'Cost' (a number input), and 'Available' (a checkbox). At the bottom of the form, there is a 'Create Product' button and a 'Back' link.



# Relaciones

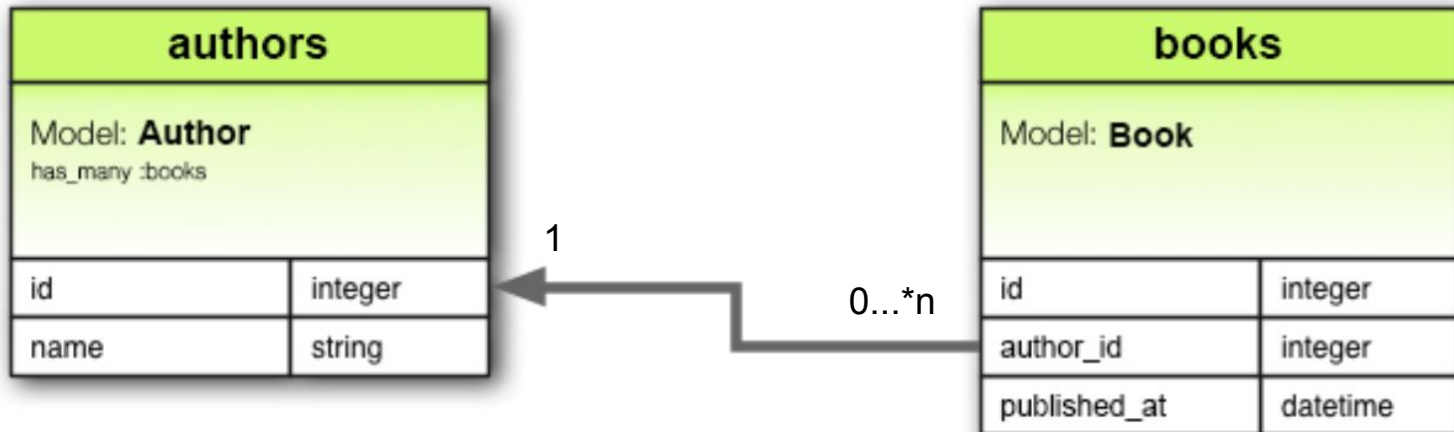


```
class Author < ApplicationRecord
  has_many :books
end
```

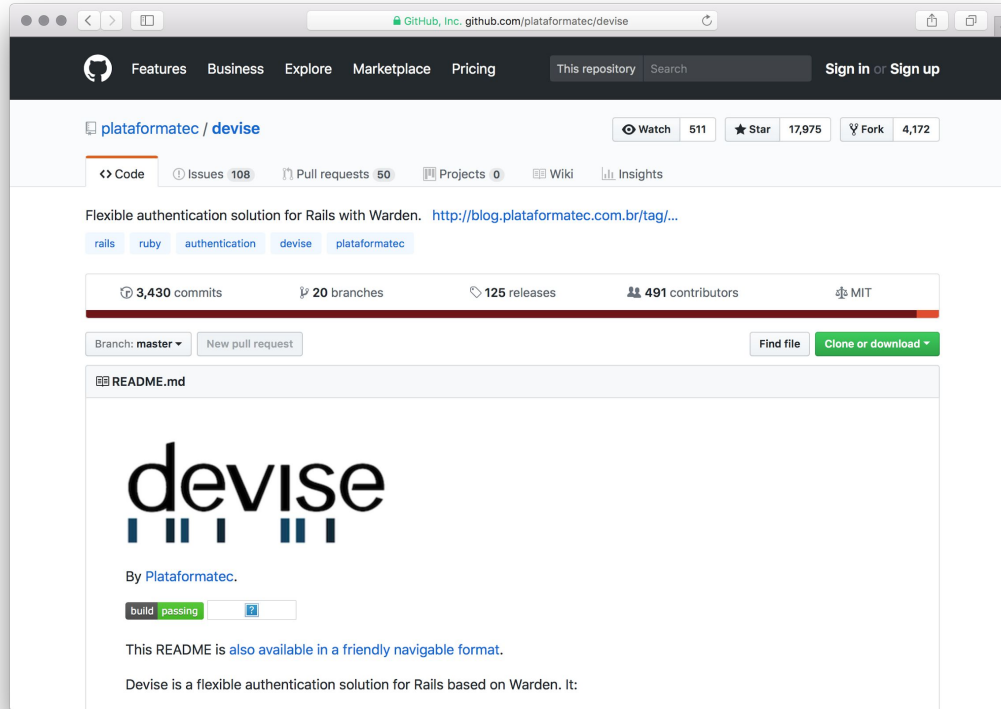
Un autor tiene muchos libros.  
Un libro tiene un autor.



```
class Book < ApplicationRecord
  belongs_to :author
end
```

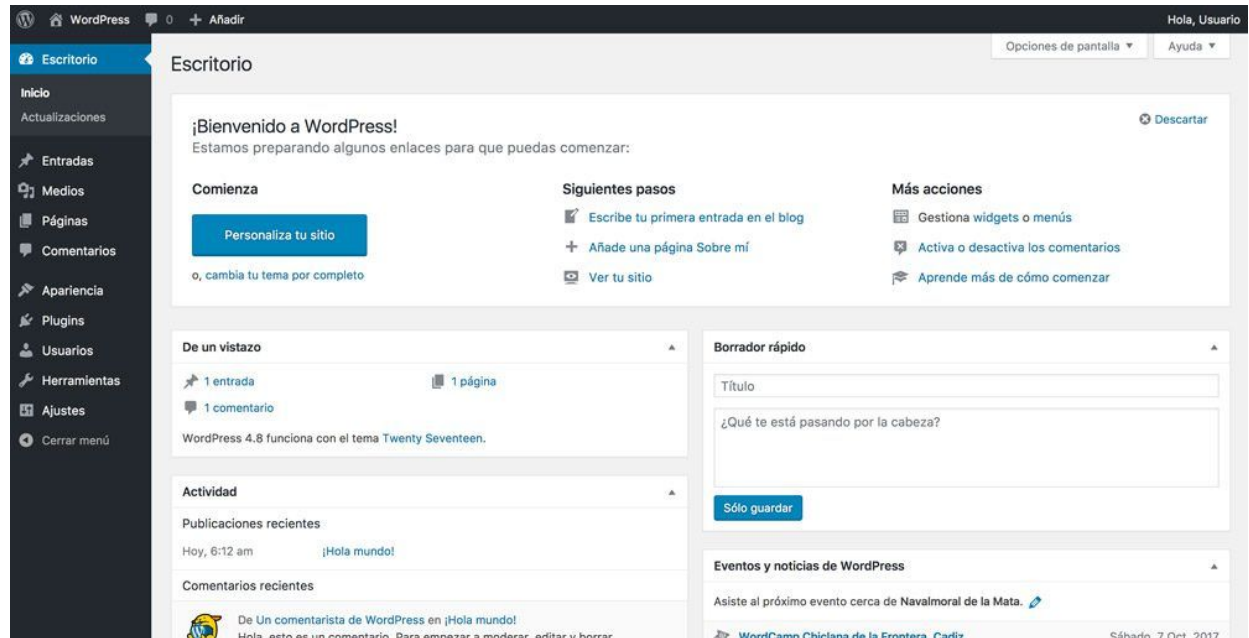


# Sesiones, Cookies y la gema Devise





# Wordpress es un CMS



**Llevar a cabo un  
proyecto TI es difícil**

—



How the customer explained it



How the Project Leader understood it



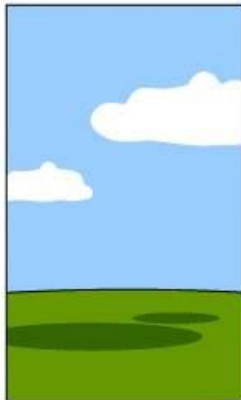
How the Analyst designed it



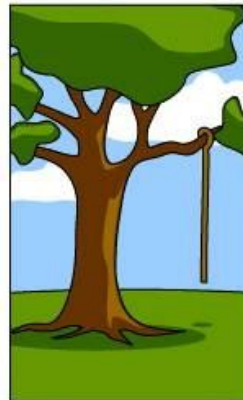
How the Programmer wrote it



How the Business Consultant described it



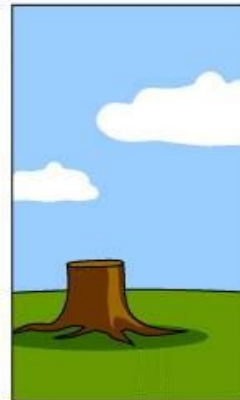
How the project was documented



What operations installed



How the customer was billed

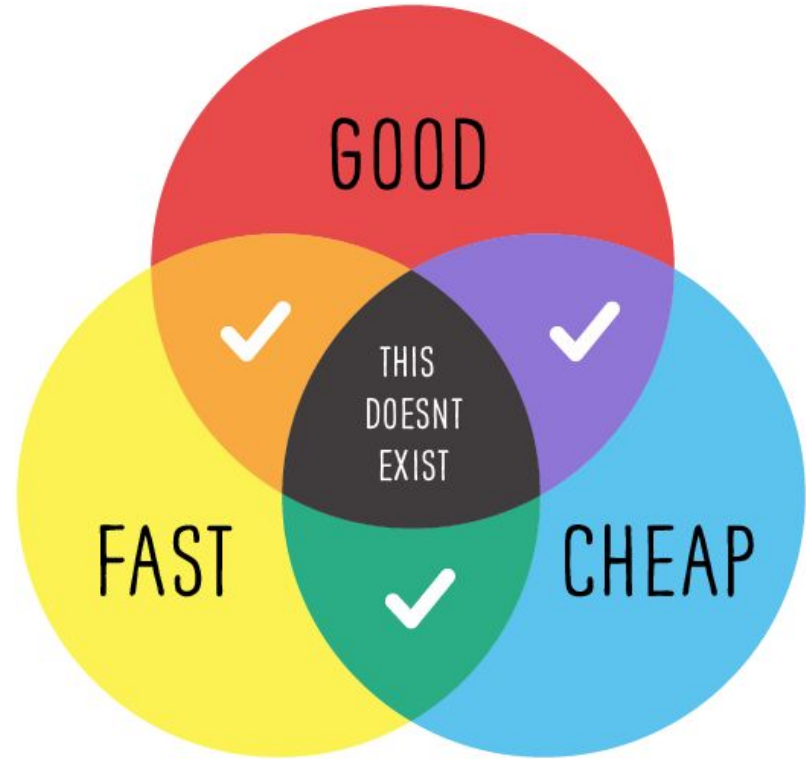


How it was supported



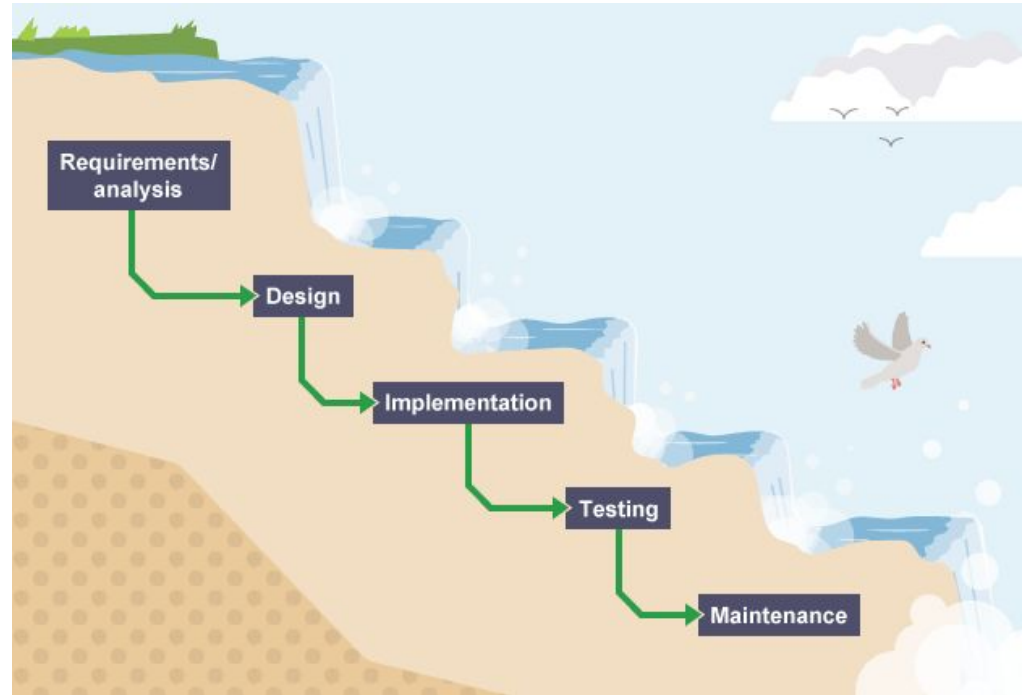
What the customer really needed

  
**No se puede tener  
todo**

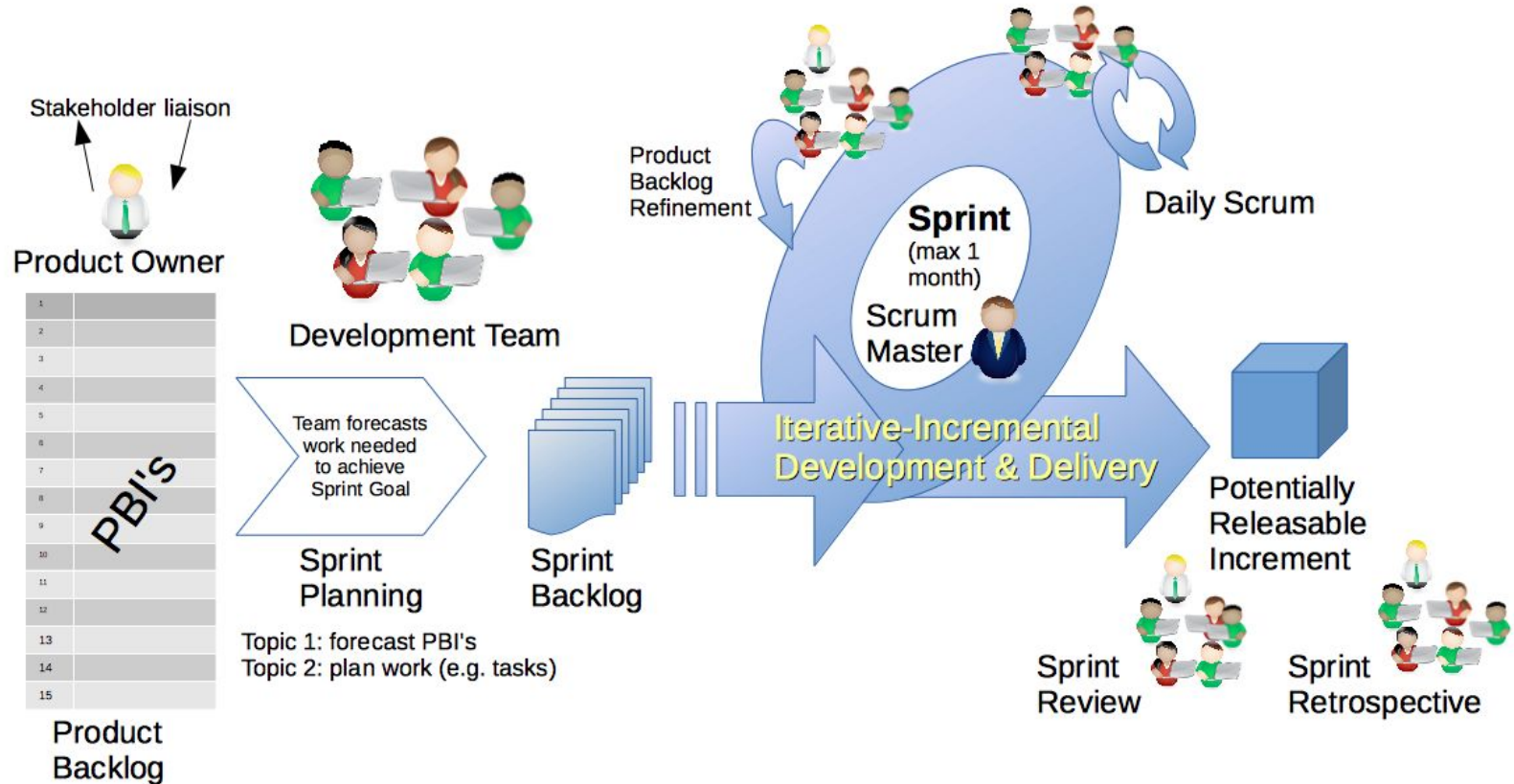


# Metodología Cascada, poco flexible.

1. Se toman los requisitos
2. Se construye el software
3. Se hacen algunas pruebas
4. Se entrega el software



# SCRUM, ágil





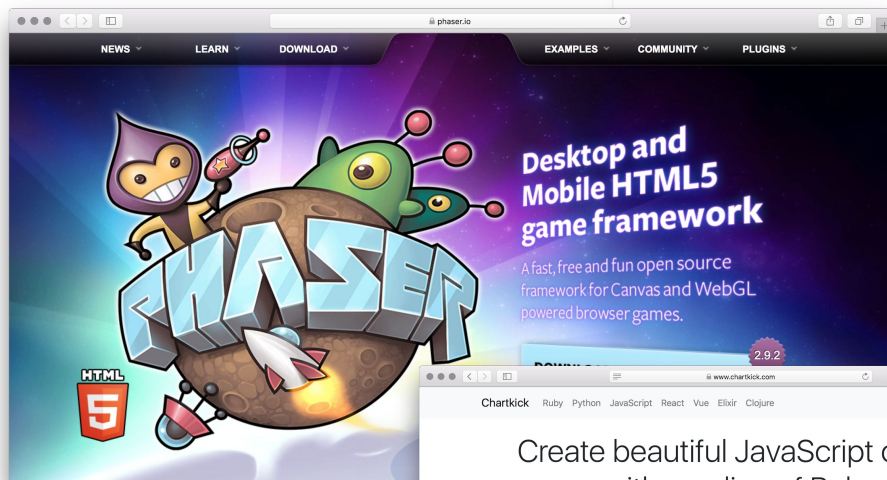
# UI / UX

User Interface / User Experience

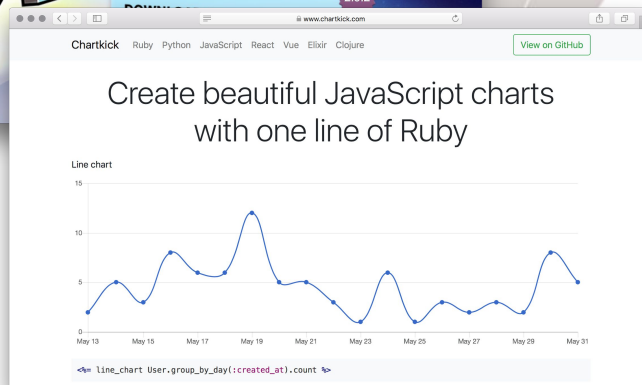


Tenemos otra herramienta más:  
**Javascript**

---



<https://phaser.io/>



<https://www.chartkick.com/>



<https://b1.ocks.org/kerryrodden/7090426>

---

¿Qué faltó ver?

---

## ¿Qué faltó ver?

- Cómo se integran SaaS y apps móviles con nuestra app en Rails
- Cómo mandar notificaciones y correos.
- Cómo lanzar la app a *producción* en Heroku

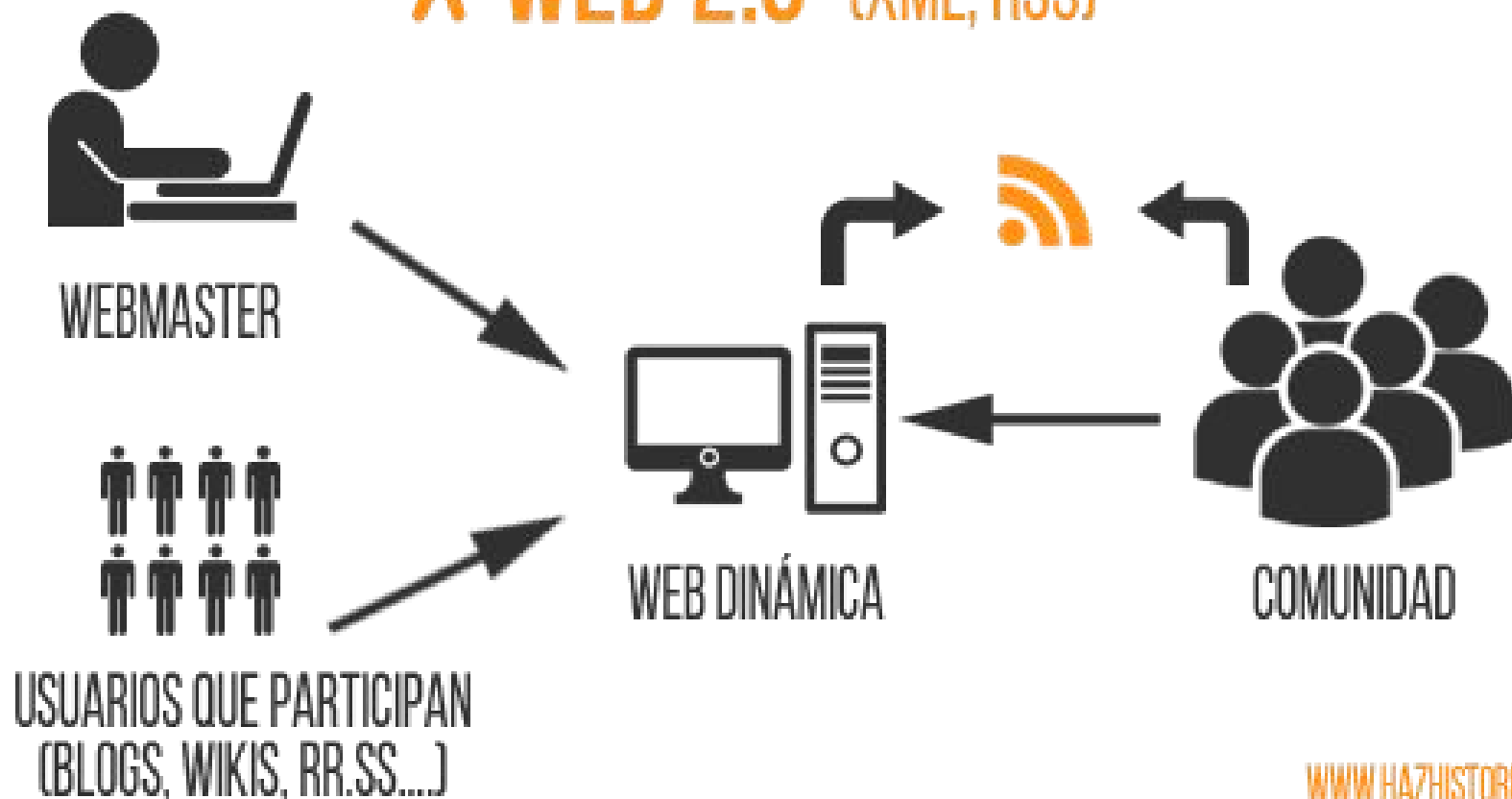
# ¿Qué se viene para el futuro?

---

# WEB 1.0 {HTML, PORTALS}

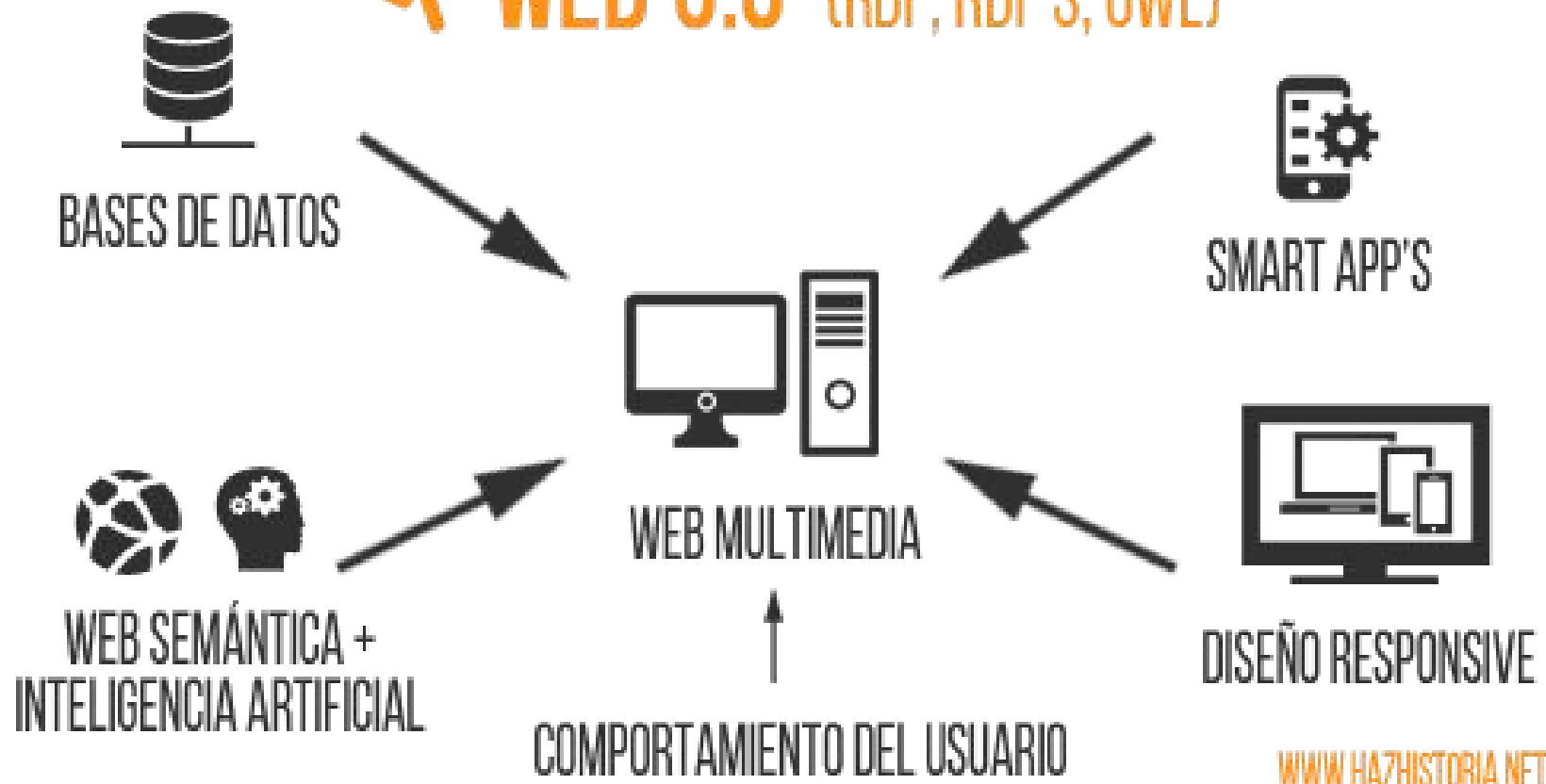


# WEB 2.0 {XML, RSS}





# WEB 3.0 {RDF, RDF'S, OWL}



# ¿WEB 4.0?

## ¿Serán las criptomonedas?





# Muchas gracias!



profesor: **Patricio López Juri** { [patricio@lopezjuri.com](mailto:patricio@lopezjuri.com) }  
créditos: **10**  
horario: **J:7-8**  
sala: **H3**



- Poner links referentes y/o mentores.