

# Presentación

**Título:** Aplicaciones Web Desconetadas  
**Autores:** Defossé Nahuel, van Haaster Diego Marcos  
**Tutor:** Saenz Lopez Marta  
**Fecha:** 15 de Diciembre 2009  
**URL:** <https://code.google.com/p/protopy>

## Objetivos Principal

"Extender un framework de aplicaciones web existente, OpenSource, de manera que una aplicación realizada sobre éste pueda ser ejecutada en el cliente de manera desconectada con un mínimo de modificaciones. Para permitir que la aplicación pueda ejecutarse en el cliente, se implementará":

- Persistencia del modelo de datos en el cliente.
- Subconjunto de acciones disponibles en modo desconectado.
- Primitivas de sincronización entre la aplicación del cliente y la aplicación web que le dio origen".

## Objetivos Secundarios

- **Open Source**

Coste de licenciamiento nulo y aseguramiento de la continuidad

- **Multiplataforma**

Windows, Linux, Mac y móviles (dode exista un browser)

- **Adaptación mínima de aplicaciones existentes**

Integración con un Frameworks Web

- **Facilidad de utilización**

Reutilizar los conceptos/patrones del framework para una rápida asimilación de los desarrolladores.

# Carencias del Browser

- Base de datos
- Servidor web
- Lenguaje de programación (consistente)
- Concurrency
- Conectividad con el entorno del cliente

## Tecnologías Existentes (1)

- **Silverlight (.NET Framework, Microsoft)**

Solución muy interesante, pero **cerrada**. *Promesa* de no cautividad.

- Otras alternativas

AIR, JavaFX, XUL, etc.

# Tecnologías Existentes

## JavaScript!

- Objetos
- Expresiones Regulares
- Patrones propios (Module, Closures)

## Librerías:

- Prototype
- Dojo
- Peppy

# Tecnologías Existentes (2)

## Google Gears

Añade al navegador 3 componentes

- **Local Server**  
Un servidor de archivos locales
- **DataBase**  
Una base de datos transaccional
- **Worker Pool**  
Sistema de Hilos con pasaje de mensajes

# JavaScript 1.7

- Generador:

Ej: Recuperación perezosa de datos.

- Orientación a objetos:

Métodos útiles como `__noSuchMethod__`, `__defineGetter__` y `__defineSetter__`.

- Azúcar sintáctico:

Asignación múltiple



# Framework

- Inversión de Control
- Comportamiento por defecto definido
- Extensibilidad
- No modificabilidad del código del framework

Todo se extiende, nada se edita

# Arquitectura

Acá va el gráfico

# Protopy

Soporte para implementación de Django sobre el navegador.

- Aprovechamiento de JavaScript 1.7
- Módulos
- OO Pythonica, `type`
- Integración de librerías:
  - API de DOM de Prototype
  - Eventos (Dojo)
  - CSS Selector (Peppy)
- **Gears**  
Enmascaramiento en el módulo `sys`, extensión de DB, Desktop.

# Arquitectura (Doff)

Ahora mostramos Doff

# Doff

Django on Protopy -> **Django Offline** :)

- API de Modelos
- Templates
- Proyecto desconectado
- **Aplicaciones adicionales**  
Sincronización Autenticación "Sesión"

# Arquitectura (Offline)

Ahora mostramos Offline

# Offline

Soporte en el servidor para proyecto desconectado

- Comandos de administración
- Instalación
- Seguridad
- Sincronización

# Demo

La aplicación de demostración es un agente de ventas.



# Conclusiones

Se lograron cumplir todos los objetivos.

# Lineas Futuras

Tomar títulos

# Miscelanea

Todo el proyecto fue realizado con software **Open Source**:

- **Firefox** (Plataforma)
- **Firebug** (Depuración, esta tesina no podría haber existido sin Firebug)
- **Django** (Framework Web)
- **Python** (Lenguaje Server Side, Scripting, Sphinx Hacking, etc.)
- **Mercurial** (control de versiones)
- **Sphinx** (para crear la documentación), LaTeX
- **rst2pdf** para crear esta presentación

Este proyecto es Open Source y se publicó en:

<http://code.google.com/p/protopy>

# FIN

¡Muchas Gracias!