

---

# **Sistemas Web Desconectados**

***Release 1***

**van Haaster, Diego Marcos; Defossé, Nahuel**

August 10, 2009



---

# Índice general

---

<b>1. Tecnologías del servidor</b>	<b>3</b>
1.1. CGI . . . . .	3
1.2. WSGI . . . . .	3
1.3. Lenguajes interpretados . . . . .	3
1.4. Frameworks web . . . . .	3
1.5. Django . . . . .	3
<b>2. Glosario</b>	<b>5</b>
<b>3. Indices, glosario y tablas</b>	<b>7</b>
<b>Índice</b>	<b>9</b>



Índice:



---

# Tecnologías del servidor

---

## 1.1 CGI

CGI es bla

## 1.2 WSGI

WSGI es CGI para Python.

## 1.3 Lenguajes interpretados

Python es un lenguaje interpretado.

## 1.4 Frameworks web

Un framework web es una cosa loca.

## 1.5 Django

[Django](#) es un framework web escrito en Python el cual sigue vagamente el concepto de Modelo Vista Controlador. Ideado inicialmente como un administrador de contenido para varios sitios de noticias, los desarrolladores encontraron que su CMS era lo suficientemente genérico como para cubrir un ámbito más amplio de aplicaciones.

En honor al músico Django Reinhardt, fue liberado el código base bajo la licencia [BSD](#) en Julio del 2005 como Django Web Framework. El slogan del framework fue “Django, El framework para perfeccionistas con fechas límites” <sup>1</sup>

---

<sup>1</sup> Del ingles “The Web framework for perfectionists with deadlines”





---

# Glosario

---

**API** [Application-Programming-Interface](#); conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

**DOM** [Document-Object-Model](#); interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos.

**JSON** [JavaScript-Object-Notation](#); formato ligero para el intercambio de datos.

**RPC** [Remote-Procedure-Call](#); es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos.

**field** An attribute on a [model](#); a given field usually maps directly to a single database column.

**generic view** A higher-order [view](#) function that abstracts common idioms and patterns found in view development and abstracts them.

**model** Models store your application's data.

**MTV** hola

**MVC** [Model-view-controller](#); a software pattern.

**project** A Python package – i.e. a directory of code – that contains all the settings for an instance of Django. This would include database configuration, Django-specific options and application-specific settings.

**property** Also known as “managed attributes”, and a feature of Python since version 2.2. From [the property documentation](#):

Properties are a neat way to implement attributes whose usage resembles attribute access, but whose implementation uses method calls. [...] You could only do this by overriding `__getattr__` and `__setattr__`; but overriding `__setattr__` slows down all attribute assignments considerably, and overriding `__getattr__` is always a bit tricky to get right. Properties let you do this painlessly, without having to override `__getattr__` or `__setattr__`.

**queryset** An object representing some set of rows to be fetched from the database.

**slug** A short label for something, containing only letters, numbers, underscores or hyphens. They're generally used in URLs. For example, in a typical blog entry URL:

`http://www.djangoproject.com/weblog/2008/apr/12/spring/`

the last bit (`spring`) is the slug.

**template** A chunk of text that separates the presentation of a document from its data.

**view** A function responsible for rendering a page.

**BSD** ve ese de

---

# Indices, glosario y tablas

---

- *Index*
- *Module Index*
- *Glosario*



---

# Índice

---

## A

API, [5](#)

## B

BSD, [6](#)

## D

DOM, [5](#)

## F

field, [5](#)

## G

generic view, [5](#)

## J

JSON, [5](#)

## M

model, [5](#)

MTV, [5](#)

MVC, [5](#)

## P

project, [5](#)

property, [5](#)

## Q

queryset, [5](#)

## R

RPC, [5](#)

## S

slug, [5](#)

## T

template, [6](#)

## V

view, [6](#)