



TECNOLÓGICO  
NACIONAL DE MÉXICO®

Instituto Tecnológico de Villahermosa

Carrera:

Ingeniería en Tecnologías de la Información y Comunicaciones

Semestre: 8vo

Materia: Introducción a las Nuevas Tecnologías de Desarrollo

Catedrático: Manuel Vergel Escamilla

Nombre de los alumnos:

Córdova de la Cruz Isaí Elías

Guevara de los Santos Dariana

Primo Escobar Herman Enrique

Santos Cano Emmanuel Urbano

Actividad

Fecha:

15 de Mayo de 2021

- Gutiérrez, Á. (2017). Desarrollo y programación en entornos web. México.
- López Sanz, M. (2016). MF0492\_3 Programación Web en el Entorno Servidor. RA-MA Editorial.
- Pérez Martínez, E. (2016). Desarrollo de aplicaciones mediante el Framework de spring.
- RA-MA Editorial. Sznajdleder, P. (2024). Java a fondo. Curso de programación. Marcombo.

Ángel Gutiérrez González • José Luis López Goytia

# DESARROLLO Y PROGRAMACIÓN EN ENTORNOS WEB



 **Alfaomega**

## 1.2 Los cimientos de un sitio web: la idea general que sustenta

¿Hay un camino único y general para entender la visión y el sentir de los usuarios? ¡Claro que no! Pero sí pueden mencionarse criterios generales aplicables a la mayor parte de las situaciones, a partir de los cuales cada organización y cada proyecto deben definir una estrategia específica.

¿Qué espera un usuario de un sitio web? Posiblemente las respuestas más comunes sean: “que funcione”, “que cubra sus necesidades”, “que le ayude a solucionar sus problemas”. Esas frases, con mucha dosis de verdad, necesitan ser analizadas a mayor profundidad para que nos lleven a criterios técnicos que debemos cubrir. Citemos un ejemplo hipotético de una situación muy común: un procesamiento “en lote” (como el cálculo de una nómina, un cierre contable o la actualización de un antivirus). El usuario no sabrá el esfuerzo que implica desarrollar un código robusto que disminuye el tiempo de procesamiento, pero sí percibirá que un software tarda diez minutos en hacer un cálculo, mientras otro lo hace en un minuto. Además, se dará cuenta de que uno no indica nada y no sabe si está corriendo, mientras el otro muestra una barra de avance. Seguramente, terminará cerrando la ventana del software de diez minutos por creer que está “trabado”, mientras que al otro lo dejará finalizar. Tal vez la reducción de tiempo implique conocimientos avanzados de programación en bases de datos o complejidad computacional, o quizá un programador pasó por alto recomendaciones básicas sobre la explotación de índices en la base de datos y por eso se aumentó el tiempo de procesamiento. En cuanto a la barra de avance, por lo común no se requieren conocimientos avanzados, pero sí pensar en el usuario y, como consecuencia, entender que es muy probable que ante la falta de mensajes termine cerrando la ventana: los conocimientos técnicos deben ir de la mano con la empatía. Por ello, antes de adentrarnos en los aspectos técnicos, habría que responder la siguiente pregunta: ¿qué esperan los usuarios de un sitio web?

### Sugerencia didáctica:

Existe un ejercicio interesante que complementa la explicación proporcionada en esta sección: tomar al menos dos sitios web que se usen mucho, ya sea para PC o celular, y tratar de contestar las siguientes preguntas: ¿cuál es la función principal que realizan?, ¿qué tareas pueden llevar a cabo?, ¿algunas están de sobra?, ¿hay funcionalidades que dejan fuera y deberían tener? También es muy ilustrativo aplicar los mismos cuestionamientos a un par de aplicaciones que no usamos porque finalmente no nos convencieron.

### 1.2.1 Los cimientos: el enfoque general del sitio web

A riesgo de ser demasiado esquemáticos, diremos que existen dos grandes enfoques para desarrollar un sitio web. El primero, que llamaremos “tradicional”, parte de las necesidades que indica un usuario y después de una etapa de análisis se procede a automatizar el proceso que ya estaba previamente sistematizado. Por lo común se aplica el método de cascada (muy recomendable para sistemas pequeños y con requerimientos que casi no cambian), aunque nada impide que los proyectos se basen en otras estrategias como los enfoques en espiral, por versiones o por prototipos, llevados a cabo a través de metodologías como Scrum, Programación Extrema o Proceso Unificado cuando se presentan desarrollos a

**CERTIFICADO DE PROFESIONALIDAD**

# **PROGRAMACIÓN WEB EN EL ENTORNO SERVIDOR**

**MF0492\_3**

**MARCOS LÓPEZ SANZ  
FRANCISCO J. SOLTERO DOMINGO  
DIANA M. SÁNCHEZ FÚQUENE  
ÁNGEL MORENO PÉREZ  
VERÓNICA A. BOLLATI  
JUAN MANUEL VARA MESA**



**Ra-Ma®**

**[www.ra-ma.es/cp](http://www.ra-ma.es/cp)**

Material protegido por derechos de autor

Para trabajar de forma profesional en un proyecto, muchas veces necesitamos algo más que una lista de tareas. Si los clientes son varios, entonces ya es imprescindible encontrar algún buen programa de gestión de proyectos, estable pero también flexible. Existen herramientas de pago como puede ser el Microsoft Project que es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo. Es útil para la gestión de proyectos, aplicando procedimientos descritos en el PMBoK del Project Management Institute.

Por otro lado podemos encontrar programas de software libre que además de ser potentes, cuentan detrás con una comunidad de desarrolladores y pueden hacernos ahorrar bastante en costes.

- Colabtive: es la alternativa *open source* a herramientas propietarias como Basecamp. Permite importar desde Basecamp e incluye funciones similares como la gestión de diferentes proyectos, los *milestones* y las listas de tareas. También mide el tiempo dedicado a las tareas, emite informes y cuenta con varios *plug-ins* para extender sus funciones.
- Project HQ: también similar a Basecamp, Project HQ está construido sobre Python, Pylons y SQLAlchemy, y su base de datos es totalmente independiente. Gestiona distintas compañías, miembros y proyectos y cuenta con *milestones* y listas de tareas. Es configurable visualmente usando CSS.
- Gantt PV: gratuito, es un programa simple, sin complicaciones, que se basa en diagramas de Gantt para planificaciones de proyectos y seguimiento de tareas. Está disponible para Windows, Mac OS y Linux.
- Clocking IT: tiene diagramas de Gantt interactivos, más otras utilidades como contador de tiempo, varias formas de comunicación, seguimiento e indexación de los cambios y unos muy buenos informes de avance.

- TeamWork: una excelente interfaz para una herramienta online que permite hacer un seguimiento de distintos proyectos y equipos de trabajo, con una versión optimizada para acceder desde móviles. Tienen licencias gratuitas para organizaciones sin ánimo de lucro y *bloggers*. Disponible para Mac OS X, Linux y Windows.
- iceScrum: tiene la misma interfaz para todos los roles. Incluye registros de historias de usuario (*backlogs*), de asuntos, de problemas y pruebas, chat en línea, *timeline* e indicadores de producto.
- Achievo: además de la utilidad de gestión de proyecto, que divide según el tiempo de su ejecución, incluye calendarios, estadísticas, plantillas y notas. No hay tarifas de licencia o limitaciones para su uso.
- dotProject.net: otra herramienta basada en la web dotProject. Lleva un tiempo funcionando y no hay ninguna empresa detrás de ella, está sostenida por voluntarios y usuarios. Permite la gestión para múltiples clientes, con herramientas para gestión de tareas, agendas y comunicaciones.
- GanttProject: un programa de escritorio multiplataforma que corre sobre Windows, Mac OS X y Linux, totalmente gratuito. Incluye diagramas de Gantt, asignación de las personas que trabajarán en el proyecto, y permite exportar los diagramas como imágenes, mientras genera informes en PDF y HTML. Permite interactuar con Microsoft Project, importando y exportándolos a sus formatos.
- TaskJuggler: un gestor de proyectos realmente potente y superior a otros que usan herramientas para editar diagramas de Gantt. Cubre todos los aspectos de desarrollo de un proyecto, desde la primera idea hasta su fin. Ayuda a medir su campo de alcance, asignación de recursos, esquema de costos y ganancias, riesgo y gestión de las comunicaciones.



# Desarrollo de aplicaciones mediante el Framework de Spring

Una panorámica del framework para J2EE  
más utilizado del momento



**Eugenia Pérez Martínez**

WWW



Desde [www.ra-ma.es](http://www.ra-ma.es) podrá  
descargar el material adicional  
con el código fuente del libro.



**Ra-Ma®**

## 1.1 INTRODUCCIÓN

---

Spring es un *framework* de desarrollo y contenedor de inversión de control *open source* para crear aplicaciones Java.

El creador de Spring es Rod Johnson, quien lo lanzó junto a la publicación de su libro *Expert One-on-One J2EE Design and Development* (Wrox Press, octubre 2002). En junio de 2003 vio la luz la primera versión del *framework* bajo la licencia Apache 2.0.

El primer gran lanzamiento fue la versión 1.0, que apareció en marzo de 2004. Al poco tiempo fue seguida por otros hitos en septiembre de 2004 y marzo de 2005. Los premios llegaron con la versión 1.2.6, cuando Spring obtuvo los reconocimientos *Jolt Awards* y *Jax Innovation Awards* en 2006. En ese mismo año fue lanzada la versión 2.0 del *framework*, posteriormente la versión 2.5 en noviembre de 2007, la 3.0 en diciembre de 2009, y dos años más tarde la 3.1. En enero de 2013 se anunció el inicio del desarrollo de la versión 4.0. Finalmente, en el momento de escribir este documento nos encontramos en la versión 4.2.0.

Una aplicación Java normalmente consiste en una serie de objetos que colaboran para conseguir el propósito de la aplicación. De ahí que los objetos de una aplicación tengan dependencia unos de los otros.

Aunque la plataforma Java proporciona una gran cantidad de funcionalidades de desarrollo de aplicaciones, tiene carencias en la manera de organizar las distintas partes de una aplicación en un todo, delegando esta tarea en arquitectos

Material protegido por derechos de autor

y desarrolladores. Los profesionales pueden apoyarse en los distintos patrones de diseño existentes (por ejemplo *Factory*, *Abstract Factory*, *Builder*, *Decorator*, o *Service Locator*), sin embargo estos patrones no son más que simplemente buenas prácticas que uno debe implementar por sí mismo.

El componente de inversión de control del *framework* Spring se encarga de solucionar esta dificultad proporcionando una manera formalizada de componer y combinar los distintos componentes de una aplicación.



# JAVA A FONDO

## CURSO DE PROGRAMACIÓN

ACTUALIZADO A JAVA 8 / INCLUYE INTRODUCCIÓN A HIBERNATE Y SPRING

**3RA. EDICIÓN**

ING. PABLO AUGUSTO SZNAJDLEDER



 **Alfaomega**

petando mayúsculas y minúsculas).

### 1.2.1 El Entorno Integrado de Desarrollo (IDE)

Si bien podemos editar nuestro código utilizando cualquier editor de texto y luego compilarlo en línea de comandos, lo recomendable es utilizar una herramienta que nos ayude en todo el proceso de programación.

Una IDE (*Integrated Development Enviroment*) es una herramienta que permite editar programas, compilarlos, depurarlos, documentarlos, ejecutarlos, etc.

Para trabajar con Java existen en el mercado diferentes IDE. Algunas son de código abierto (*open source*) como *Eclipse* y *NetBeans* y otras son pagas e impulsadas por las empresas de tecnología como *JBuilder* (de *Borland*), *JDeveloper* (de *Oracle*), *WebPhere* (de *IBM*), etcétera.



Instalar Java y *Eclipse*.



Crear y ejecutar nuestro primer programa en *Eclipse*.

En este libro utilizaremos *Eclipse* que, además de ser gratis, es (para mi gusto) la mejor herramienta de programación de todos los tiempos.

Para no “ensuciar” el contenido del capítulo con una excesiva cantidad de imágenes y contenidos básicos que (probablemente) para el lector podrían resultar triviales se proveen videotutoriales en los que se explica en detalle cómo instalar Java y *Eclipse* y cómo utilizar *Eclipse* para escribir y ejecutar programas Java. Si el lector no tiene conocimientos previos de programación se provee también un apéndice donde se explican las bases necesarias para que pueda comprender los contenidos expuestos en este libro.

- Documentación oficial de Django – <https://docs.djangoproject.com/en/stable/> Fuente principal con guías, tutoriales y referencias técnicas.
- Django Project (Sitio oficial) – <https://www.djangoproject.com/> Noticias, comunidad y recursos sobre Django.
- Real Python - Django Tutorials – <https://realpython.com/tutorials/django/> Blog con guías y tutoriales para aprender Django paso a paso.
- MDN Web Docs - Django Introduction – <https://developer.mozilla.org/enUS/docs/Learn/Server-side/Django> Introducción al framework en la documentación de Mozilla.
- Django for Beginners (Libro) – William S. Vincent Un libro recomendado para principiantes que quieren aprender a desarrollar aplicaciones con Django.

## The model layer 📄

Django provides an abstraction layer (the “models”) for structuring and manipulating the data of your web application. Learn more about it below:

- **Models:** [Introduction to models](#) | [Field types](#) | [Indexes](#) | [Meta options](#) | [Model class](#)
- **QuerySets:** [Making queries](#) | [QuerySet method reference](#) | [Lookup expressions](#)
- **Model instances:** [Instance methods](#) | [Accessing related objects](#)
- **Migrations:** [Introduction to Migrations](#) | [Operations reference](#) | [SchemaEditor](#) | [Writing migrations](#)
- **Advanced:** [Managers](#) | [Raw SQL](#) | [Transactions](#) | [Aggregation](#) | [Search](#) | [Custom fields](#) | [Multiple databases](#) | [Custom lookups](#) | [Query Expressions](#) | [Conditional Expressions](#) | [Database Functions](#)
- **Other:** [Supported databases](#) | [Legacy databases](#) | [Providing initial data](#) | [Optimize database access](#) | [PostgreSQL specific features](#)

## The view layer 📄

Django has the concept of “views” to encapsulate the logic responsible for processing a user’s request and for returning the response. Find all you need to know about views via the links below:

- **The basics:** [URLconfs](#) | [View functions](#) | [Shortcuts](#) | [Decorators](#) | [Asynchronous Support](#)
- **Reference:** [Built-in Views](#) | [Request/response objects](#) | [TemplateResponse objects](#)
- **File uploads:** [Overview](#) | [File objects](#) | [Storage API](#) | [Managing files](#) | [Custom storage](#)
- **Class-based views:** [Overview](#) | [Built-in display views](#) | [Built-in editing views](#) | [Using mixins](#) | [API reference](#) | [Flattened index](#)
- **Advanced:** [Generating CSV](#) | [Generating PDF](#)
- **Middleware:** [Overview](#) | [Built-in middleware classes](#)

## The template layer 📄

The template layer provides a designer-friendly syntax for rendering the information to be presented to the user. Learn how this syntax can be used by designers and how it can be extended by programmers:

- **The basics:** [Overview](#)
- **For designers:** [Language overview](#) | [Built-in tags and filters](#) | [Humanization](#)
- **For programmers:** [Template API](#) | [Custom tags and filters](#) | [Custom template backend](#)

## Forms 📄

Django provides a rich framework to facilitate the creation of forms and the manipulation of form data.

- **The basics:** [Overview](#) | [Form API](#) | [Built-in fields](#) | [Built-in widgets](#)
- **Advanced:** [Forms for models](#) | [Integrating media](#) | [Formsets](#) | [Customizing validation](#)

# Meet Django

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.



## Ridiculously fast.

Django was designed to help developers take applications from concept to completion as quickly as possible.



## Reassuringly secure.

Django takes security seriously and helps developers avoid many common security mistakes.



## Exceedingly scalable.

Some of the busiest sites on the web leverage Django's ability to quickly and flexibly scale.

[Learn more about Django](#)



# Django Web Framework (Python)

[Overview: Django Web Framework \(Python\)](#)[Next](#)

Django is an extremely popular and fully featured server-side web framework, written in Python. This module shows you why Django is one of the most popular web server frameworks, how to set up a development environment, and how to start using it to create your own web applications.

## Prerequisites

Before starting this module you don't need to have any knowledge of Django. Ideally, you would need to understand what server-side web programming and web frameworks are by reading the topics in our [Server-side website programming first steps](#) module.

A general knowledge of programming concepts and [Python](#) is recommended, but is not essential to understanding the core concepts.

**Note:** Python is one of the easiest programming languages for novices to read and understand. That said, if you want to understand this module better, there are numerous free books and tutorials available on the internet to help you out (new programmers might want to check out the [Python for Non Programmers](#) page on the python.org wiki).

## Tutorials

### [Django introduction](#)

In this first Django article we answer the question "What is Django?" and give you an overview of what makes this web framework special. We'll outline the main features, including some advanced functionality that we won't have time to cover in detail in this module. We'll also show you some of the main building blocks of a Django application, to give you an idea of what it can do before you

A pink five-pointed star with the number 4.2 inside it.

4.2

# DJANGO

— *for* —

# BEGINNERS

Build websites with Python & Django

WILLIAM S. VINCENT