



**UNIVERSIDAD POLITÉCNICA DE MADRID**

Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

**MÁSTER EN INGENIERÍA WEB**

**Proyecto Fin de Máster**

---

Módulo de Drupal para Generación de PDF con Ubercart

**Autor**

**Jorge Cárdenas Caballero**

**Tutor**

**Eduardo García Pardo**

22 de junio de 2017

# Agradecimientos

Agradezco toda la ayuda prestada por ...



# Resumen

En la actualidad el mundo del comercio electrónico está viviendo una edad de oro. Empresas como Amazon obtienen al año más de 100.000 millones en ventas on-line con alrededor de 300 millones de clientes en todo el mundo.

Uno de los problemas de las plataformas de e-commerce actuales, radica en que un cliente necesita en ocasiones guardar información sobre un producto para su posterior consulta, y, por lo general, la presentación de éstos productos una vez impresos o guardados en un documento, como por ejemplo PDF, no es la más adecuada. Un administrador de la plataforma, no suele contar con medios ni formación para poder modificar la presentación de sus productos a la hora de generar un documento, teniendo que recurrir a programadores especializados que realicen la maquetación de los mismos.

Como respuesta a este problema, se propone realizar un módulo para una plataforma de e-commerce conocida con el que un administrador pueda crear plantillas de documentos PDF de forma rápida y sencilla de modo que un usuario pueda guardar la información de los productos en PDF de acuerdo a las mismas.

## PALABRAS CLAVE

E-commerce, PDF, Módulo, Plantilla



# Abstract

Nowadays the world of ecommerce is living its own golden age. Companies such as Amazon are reporting net incomes around the 100.000 million dollar value in online sales, with 300 million customers around the globe.

One of the gaps of these online platforms is the inability to allow customers to save information of a product they're interested in, in order to subsequently consult it. Currently and generally speaking, once the details of a product have been printed or saved in a document in PDF format, for instance, the information is not rendered in the most arranged way. This is normally due to the platform administrator lacking an easy means to modify how the appearance of the information will be stored, and this results in them having a dependancy on programming experts in order to provide more user friendly layouts.

As a way to mend this handicap and enhance users' experience while shopping online, the objective of this project is to develop an independant module that can easily be plugged into an ecommerce platform and allow its administrators the possibility of designing templates for the rendering of their catalog. With this functionality, users may be able to easily save the details of a product and later be able to examine it.

## KEYWORDS

E-commerce, PDF, Module, Template



# Índice general

<b>Agradecimientos</b>	<b>I</b>
<b>Resumen</b>	<b>III</b>
<b>Abstract</b>	<b>v</b>
<b>1. Objetivos</b>	<b>1</b>
1.1. Objetivo general . . . . .	1
1.2. Objetivos específicos . . . . .	1
1.2.1. Stack tecnológico . . . . .	1
1.2.2. Framework base . . . . .	1
1.2.3. Plataforma de e-commerce . . . . .	2
1.2.4. Tecnología Front-End . . . . .	2
1.2.5. Generación de PDF . . . . .	2
1.2.6. Comunicaciones Back-End -¿Front-End . . . . .	2
1.2.7. Capacidades del administrador . . . . .	2
1.2.8. Capacidades del usuario . . . . .	3
1.2.9. Empaquetado de la solución . . . . .	3
<b>2. Introducción</b>	<b>5</b>
2.1. Motivación . . . . .	6
2.2. Propuesta . . . . .	7
2.2.1. Requisitos funcionales . . . . .	7
2.2.2. Requisitos no funcionales . . . . .	8
2.3. Tecnologías . . . . .	8
2.3.1. Lenguajes . . . . .	8
2.3.2. Frameworks y librerías . . . . .	14
2.3.3. Sistemas . . . . .	17
2.3.4. Herramientas de desarrollo . . . . .	19





<b>3. Metodologías de desarrollo software</b>	<b>23</b>
3.1. ¿Por qué usar metodologías? . . . . .	23
3.2. Metodologías tradicionales vs ágiles . . . . .	24
<b>4. Descripción Informática (20-35 %)</b>	<b>27</b>
4.1. La base de datos coleccionada (si tiene sentido). . . . .	27
4.2. Los algoritmos para el desarrollo de la solución . . . . .	27
<b>5. Conclusiones y trabajos futuros (5 %)</b>	<b>29</b>
5.1. Implementación . . . . .	29
5.2. Resultados . . . . .	29
<b>Bibliografía</b>	<b>31</b>
<b>Anexo 1</b>	<b>33</b>

# Índice de figuras

2.1. Logo de HTML 5 . . . . .	9
2.2. Logo de CSS 3 . . . . .	10
2.3. Logo de Javascript . . . . .	11
2.4. Logo de PHP . . . . .	12
2.5. Logo de JSON . . . . .	13
2.6. Logo de Drupal . . . . .	14
2.7. Logo de Angularjs . . . . .	15
2.8. Logo de JQuery . . . . .	16
2.9. Logo de Apache HTTP Server . . . . .	18
2.10. Logo de MySQL . . . . .	18
2.11. Logo de Brackets . . . . .	19
2.12. Logo de Git . . . . .	20
2.13. Logo de Bower . . . . .	21



# Capítulo 1

## Objetivos

A continuación se explicarán de forma general y detallada los objetivos propuestos para este proyecto fin de máster.

### 1.1. Objetivo general

El objetivo general del proyecto, consiste en el diseño e implementación de una solución que permita la generación de documentos PDF a partir de la información obtenida de una plataforma de e-commerce. Un administrador de página debe ser capaz de gestionar la disposición de los elementos del PDF a generar de una forma ágil y sencilla. Todo este proceso será transparente para un usuario final, que únicamente con presionar un botón obtendrá la información de los productos en un documento.

### 1.2. Objetivos específicos

A partir de los objetivos generales se extraen los siguientes objetivos específicos:

#### 1.2.1. Stack tecnológico

El proyecto será realizado utilizando tecnologías de libre distribución. En concreto se utilizará un stack WAMP, acrónimo de Windows como sistema operativo, Apache como servidor web, MySQL como base de datos, PHP como lenguaje en el que se programan las aplicaciones web. Se podría haber optado también por un stack LAMP, con Linux como sistema operativo, pero ya que la selección de SO no afecta al resultado final del proyecto, se decide utilizar Windows.

#### 1.2.2. Framework base

El proyecto deberá realizarse utilizando Drupal como framework base. Drupal es un CMS, es decir, un sistema de gestión de contenidos (Content Management System) muy



utilizado actualmente para la creación de aplicaciones web. Este sistema tiene una base de usuarios muy alta, lo que asegura una gran disponibilidad de documentación y soporte. Al ser open source, los usuarios pueden expandir la funcionalidad del mismo mediante módulos funcionales. En concreto son interesantes sus módulos de e-commerce.

### **1.2.3. Plataforma de e-commerce**

Drupal tiene a su disposición varios módulos que expanden su funcionalidad para satisfacer las necesidades de los propietarios de tiendas on-line. Los más conocidos son Drupal commerce y Ubercart. El objetivo del proyecto es utilizar Ubercart ya que actualmente tiene más usuarios y una base funcional más sólida, al llevar más tiempo en el mercado.

### **1.2.4. Tecnología Front-End**

Drupal viene integrado con JQuery, que durante años fue el referente en desarrollo de páginas web dinámicas, sin embargo hoy en día está siendo sustituido por nuevos frameworks de creación web como Reactjs y Angularjs. Se pide como objetivo el uso de Angularjs para toda la parte de la administración de plantillas para generación de PDF.

### **1.2.5. Generación de PDF**

Se pide que la generación de PDF sea realizada en el lado del cliente para evitar mayores sobrecargas del servidor. Por lo tanto se utilizará una librería Javascript que permita esta generación. La librería escogida es jsPDF ya que permite generar documentos tanto en modo imagen como en modo texto, además de permitir un mayor control del posicionado de elementos en un pdf.

### **1.2.6. Comunicaciones Back-End -¿Front-End**

Para las comunicaciones de los sistemas de Front-End y Back-End se utilizará JSON ya que es un lenguaje ligero que Javascript entiende sin necesidad de transformaciones y nos permitirá un tratamiento de la información más ágil.

### **1.2.7. Capacidades del administrador**

Un administrador de la web debe ser capaz de gestionar las plantillas PDF que serán usadas para generar los documentos. Deberá poder crear, borrar, editar y activar. A la hora de editar una plantilla, deberá ser capaz de seleccionar dónde quiere posicionar cada elemento (título del producto, precio, imagen...), además de establecer sus dimensiones

y su tamaño de letra. La funcionalidad de activar una plantilla servirá para elegir cual plantilla debe ser usada para la generación de PDF.

### **1.2.8. Capacidades del usuario**

Un usuario debe ser capaz de obtener el PDF de un producto con tan sólo pulsar un botón. Este PDF se generará de acuerdo con la plantilla escogida por el administrador.

### **1.2.9. Empaquetado de la solución**

La solución debe ser exportable a otras instalaciones de Drupal que cumplan los requisitos mínimos para su instalación (Drupal 7, Ubercart, y otras dependencias). Por lo tanto, se pide la realización de un módulo de Drupal que contenga todo lo necesario para su correcto funcionamiento.



## Capítulo 2

### Introducción

El comercio electrónico tal y como lo conocemos hoy, consiste en la compra y venta de productos o servicios a través de medios electrónicos, como por ejemplo Internet. Con la llegada en masa de la red de redes a los hogares del mundo, se creó un nicho de mercado que en ese entonces estaba desaprovechado, como es el negocio de la venta on-line. Aunque anteriormente ya existían otros sistemas de e-commerce, basados en televisión, teléfono o incluso a través de Internet, no fue hasta la llegada de Amazon e EBay que el comercio on-line empezó a masificarse. A partir de este punto clave, y viendo el éxito que supone para un comercio satisfacer la demanda creciente del “cliente” surgieron cada vez más plataformas que ofertaban productos y servicios por Internet, como Netflix (Alquiler de DVDs con un sistema de reparto a domicilio), Alibaba.com (parecido a Amazon pero para el mercado chino), Zappos (plataforma de venta de zapatos) entre otros. También surgieron alrededor del e-commerce, otros servicios útiles para usuarios y vendedores, como Paypal (plataforma de pago on-line) o AdWords de Google, para mostrar publicidad relevante para el usuario.

Al principio sólo las grandes empresas podían permitirse aumentar su visibilidad on-line mediante el uso de caras soluciones personalizadas, sin embargo, poco a poco, fueron surgiendo plataformas mucho más asequibles para el pequeño comercio como pueden ser osCommerce, Magento, o PrestaShop, o Shopify, además de soluciones gratuitas basadas en frameworks de software libre como Drupal, Joomla o Wordpress.

A partir de este punto, es raro ver una tienda que no tenga opción de venta on-line, actualmente la única diferencia es el tiempo que tarda un cliente en recibir el producto, siendo las grandes aquellas que se pueden permitir envíos en un único día.

Prácticamente la totalidad de las tiendas on-line se basan en los mismos principios: se presenta un catálogo de productos con imagen, descripción y precio y el usuario puede





elegir tranquilamente y desde su hogar los elementos que va a adquirir. Actualmente, un sistema de ranking y comentarios pueden ayudar al usuario final a elegir un producto, siendo los de más valoración los que más posiblemente satisfarán sus necesidades.

En ocasiones un usuario necesita guardar la información de un producto para su consulta posterior, o un comercial necesita ofrecer a un posible comprador un catálogo personalizado de productos. En estos casos existe una limitación muy grande a la hora de generar el documento con un formato atractivo y por lo general los resultados suelen ser muy pobres y no muestran la información más relevante como al vendedor le gustaría verla, por ejemplo el precio más grande, la foto al centro etc... Todos estos retoques se tienen que pensar previamente cuando se planifica el sistema de venta on-line y, por lo general, tienen que ser realizados por un experto en la materia que no todos los comercios poseen.

## 2.1. Motivación

Como se ha explicado al final de la introducción, la motivación principal es la carencia de soluciones disponibles a la hora de elegir la disposición de elementos en un documento PDF, orientado en este caso al mundo del e-commerce.

Existen muchas soluciones de generación de PDFs, tanto a nivel de Back-End como de Front-End, con un alto nivel de personalización, pero son librerías que requieren conocimientos avanzados de programación y administración de sistemas para ser usadas, y estos conocimientos no suelen estar al alcance del propietario de una empresa que solamente quiere ofrecer sus productos en Internet.

También existen soluciones estándar, Chrome, por ejemplo nos permite convertir una página web a PDF, pero está limitado por la calidad del trabajo que el maquetador web haya realizado con el archivo print.css, en el que se guardan las reglas de estilo a la hora de imprimir las diferentes páginas de la web. De nuevo, la modificación de este archivo requiere conocimientos avanzados de maquetación web y no suele ser fácil conseguir que una página se imprima como de la forma deseada, además de que habría que ocultar elementos sobrantes, como publicidad y menús.

Por lo tanto para facilitar la labor de propietarios y administradores en la presentación de sus productos a la hora de generar un documento PDF, se propone la creación de una

aplicación que brinde la funcionalidad deseada para una plataforma de e-commerce.

## 2.2. Propuesta

A partir de lo elaborado en la introducción, motivación y objetivos, se propone por tanto la realización de un módulo para una de las plataformas libres de e-commerce más utilizadas como es Ubercart por Drupal que permita a un administrador la gestión de plantillas para generación de PDFs de productos del catálogo que el comercio tenga a la venta.

### 2.2.1. Requisitos funcionales

El módulo deberá implementar la siguiente funcionalidad

#### **Botón de generación de PDF**

Un usuario debe ser capaz de generar un documento PDF de un producto con tan sólo pulsar un botón.

#### **Administración de plantillas**

Un administrador puede acceder a una sección de administración de plantillas, protegida por privilegios especiales.

#### **Creación de plantillas**

Un administrador puede crear plantillas nuevas proporcionando un nombre de plantilla y un tipo.

#### **Borrado de plantillas**

Un administrador puede borrar plantillas antiguas para no sobrecargar la base de datos.

#### **Activado de plantillas**

Un administrador puede activar una plantilla para ser usada en la generación de PDFs.

#### **Editado de plantillas**

Un administrador puede editar las plantillas entrando en una pantalla de edición en la que se muestran diferentes opciones.



### **Mover elementos de una plantilla**

Un administrador en modo de edición de plantilla puede mover los elementos y posicionarlos dónde desee.

### **Cambiar tamaño de elementos**

Un administrador en modo de edición de plantilla puede cambiar el tamaño de los elementos de la misma.

### **Cambiar tamaño de fuente**

Un administrador en modo de edición de plantilla puede cambiar el tamaño de fuente de los elementos de la misma.

### **Botón de guardar cambios de plantilla**

Un administrador en modo de edición de plantilla puede pulsar un botón para guardar los cambios de la misma.

## **2.2.2. Requisitos no funcionales**

El módulo deberá cumplir los siguientes requisitos no funcionales

### **Uso de tecnologías de libre distribución**

Las tecnologías usadas para el desarrollo del módulo deben ser de libre distribución.

### **Módulo exportable**

El módulo desarrollado debe poderse instalar en otras instalaciones de Drupal que cumplan los requisitos y dependencias.

## **2.3. Tecnologías**

En este capítulo se enumerarán las tecnologías que se han utilizado para la realización de éste proyecto.

### **2.3.1. Lenguajes**

Los lenguajes de programación usados para el desarrollo del proyecto aparecen explicados a continuación.

#### **HTML**



**Figura 2.1:** Logo de HTML 5

El lenguaje Hypertext Markup Language ó HTML sirve principalmente para la publicación y estructuración de documentos y especificación de hipervínculos. Junto a CSS y Javascript, forman la base de la programación web actual. Es un lenguaje basado en etiquetas y todo el contenido de un documento está almacenado entre una etiqueta de apertura y otra de cierre. La primera descripción publica de HTML fue un documento llamado 'HTML Tags' publicado por Tim Berners-Lee a finales de 1991 y describe los primeros 18 elementos incluidos en el lenguaje. Durante todo este tiempo se han ido publicando nuevas versiones de HTML en las que se añadían nuevas etiquetas y se eliminaban otras obsoletas. La versión actual es HTML5 que añade múltiples etiquetas avanzadas para controles multimedia, validaciones de formularios y maquetación de páginas web optimizadas para buscadores.

Para programar en HTML basta con usar un editor de texto cualquiera y un navegador web para su renderizado. La extensión de un documento HTML es '.html' o '.htm'.

Ejemplo:

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>
    <p>¡Hola, Mundo!</p>
  </body>
</html>
```

CSS



**Figura 2.2:** Logo de CSS 3

CSS es el acrónimo de Cascade Style Sheets el cuál es un lenguaje de estilo usado para describir la presentación de un lenguaje basado en etiquetas o marcas como HTML. Su diseño está enfocado principalmente a separar la presentación del contenido del documento, incluyendo aspectos como la estructura, colores y fuentes. Las especificaciones CSS son mantenidas por el World Wide Web Consortium (W3C). La primera propuesta de CSS la realizó Håkon Wium Lie en 1994 y a pesar de que en el momento habían más propuestas de lenguajes de estilo, CSS se impuso y acabó siendo lanzado en 1996 por el W3C. El lenguaje tuvo unos inicios complicados ya que los navegadores inicialmente ofrecían soporte muy limitado para CSS. No fue sino hasta la versión 5 de Internet Explorer que se dio soporte a la totalidad del lenguaje, y aun así muchos otros navegadores seguían ofreciendo compatibilidades limitadas. Actualmente la especificación CSS utilizada es la 3, aunque a partir de esta versión, el lenguaje se partió en módulos y podemos hacer uso de algunos de la especificación 4.

Al igual que con HTML, para programar en CSS basta con un programa editor de texto y un navegador web. Es un lenguaje ligado a HTML y puede aparecer o bien incrustado en el mismo documento '.html' o bien en su propio archivo '.css' que luego es importado al documento web.

Ejemplo:

```
body {  
    font-size: 12px;  
}  
div {  
    color: #fff;  
}
```

**Javascript**



**Figura 2.3:** Logo de Javascript

Javascript es un lenguaje de programación de alto nivel creado originalmente en el año 1995 para ser un lenguaje de scripting complementario a los navegadores web. Inicialmente se le dio el nombre de Mocha en su fase de desarrollo, aunque luego su nombre cambió a LiveScript para más tarde ser reemplazado por Javascript en su lanzamiento junto a Netscape Navigator 2.0. Se cree que se le dio este nombre para aprovechar la popularidad de Java. Un año después fue adoptado por Microsoft bajo el nombre de JScript para evitar problemas legales de nomenclatura y más tarde fue estandarizado por ECMA Internacional lanzando la especificación de lenguaje ECMAScript. Las especificaciones de ECMAScript se han ido adaptando a los tiempos modernos, añadiendo nuevas funcionalidades y corrigiendo otras. Actualmente se trabaja mucho con la especificación ECMAScript 5, aunque la especificación 6 está ganando popularidad desde su salida en el año 2015 y sobre todo por la adopción de la misma por los desarrolladores de los frameworks y librerías más populares como Angularjs, Reactjs y Babel.

Javascript es un lenguaje dinámico, sin tipado e interpretado en tiempo de ejecución. Es un lenguaje con múltiples paradigmas: funcional, imperativo e incluso en las últimas especificaciones, integra orientación a objetos.

Para programar en Javascript, basta con usar un editor de texto y un navegador web, aunque para aprovechar todas sus funcionalidades, se requiere de un servidor web configurado adecuadamente, ya que, por ejemplo el acceso a archivos de la máquina podría darnos problemas. Es posible incrustar scripts de Javascript en un documento HTML o bien desarrollar en un documento aparte e incrustarlo posteriormente. La extensión de Javascript es '.js'.

Ejemplo:

```
function hello() {
  console.log("!Hola Mundo!");
}
```

## PHP



**Figura 2.4:** Logo de PHP

PHP es un lenguaje de scripting para Back-End diseñado principalmente para diseño web, pero que es utilizado también como lenguaje de propósito general. Fue creado en el año 1995 por Rasmus Lerdorf quien escribió unas aplicaciones de consola en C para mantener su página web personal. Posteriormente las amplió para funcionar con WebForms y comunicarlas con bases de datos y lo llamó "Personal Home Page/Forms Interpreter."° PHP/FI. Inicialmente no fue concebido para ser un lenguaje de programación, pero Lerdorf fue paulatinamente añadiendo funcionalidades hasta que se formó un equipo de desarrollo, lanzando su primera versión oficial en 1997.

Durante su tiempo de vida, el lenguaje ha recibido numerosas actualizaciones. Actualmente nos encontramos en la séptima iteración de PHP, con la salida de la versión 7.2 programada para noviembre de 2017.

Es un lenguaje orientado objetos pero que nos permite utilizar varios paradigmas de programación como el estructurado, imperativo, funcional y reflexivo. Como suele ocurrir con los lenguajes de scripting, PHP es un lenguaje interpretado, es decir, no requiere de compilación previa para su ejecución. Además es de tipado dinámico, por lo que las variables no están asociadas a un tipo de datos en el momento de su declaración, si no que se le asigna un tipo automáticamente en el momento de su uso.

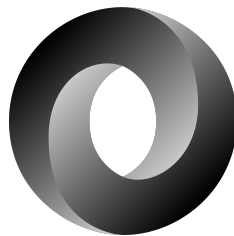
Para empezar a programar en PHP se requiere un servidor web capacitado para su ejecución y además los paquetes de comandos de consola requeridos para el inicio de programas escritos en este lenguaje. Un servidor muy utilizado es Apache. Escribir programas en PHP es sencillo y la forma más rápida consiste en incrustar pequeños scripts en los documentos HTML de nuestra web que serán interpretados por el servidor y completamente invisibles para un usuario final. Los ficheros de lenguaje PHP tienen la extensión '.php'.

Ejemplo:

```
<!DOCTYPE HTML>
```

```
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>
    <?php
      echo "¡Hola, Mundo!";
    ?>
  </body>
</html>
```

## JSON



**Figura 2.5:** Logo de JSON

JSON es el acrónimo de JavaScript Object Notation y más que un lenguaje de programación, se trata de un estándar abierto de formato de archivo para el envío de información a través de pares clave-valor. Es muy utilizado hoy en día para las comunicaciones entre cliente y servidor. La primera especificación de JSON vino de la mano de Douglas Crockford quien lo creó originariamente para ser un subconjunto de Javascript. De hecho es actualmente muy utilizado junto a este lenguaje, sin embargo JSON es independiente del lenguaje con el que sea usado. En el año 2013 se estableció un estándar por ECMA Internacional.

Los tipos de datos que pueden ser usados en un documento JSON son: Boolean, String, Number, Array, Object y null.

Ejemplo:

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {
```



```
"streetAddress": "21 2nd Street",  
"city": "New York"  
},  
"phoneNumbers": [  
  {  
    "type": "home",  
    "number": "212 555-1234"  
  },  
  {  
    "type": "office",  
    "number": "646 555-4567"  
  }  
],  
"spouse": null  
}
```

### 2.3.2. Frameworks y librerías

En este capítulo se hace una recopilación de los frameworks y librerías de terceros usados durante el desarrollo del proyecto y sus características.

#### Drupal



**Figura 2.6:** Logo de Drupal

Drupal es un sistema de gestión de contenidos (o CMS según sus siglas en inglés) de código abierto que es utilizado por un 2,3 % de sitios web del mundo. Es un sistema libre, modular y multipropósito que permite publicar artículos, imágenes, archivos y que también ofrece la posibilidad de otros servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos. Es un sistema de carga dinámico, es decir, todos los recursos se almacenan en una base de datos y se cargan en tiempo real a la hora de generar una página. Este CMS está escrito utilizando el lenguaje PHP valiéndose de las librerías de Symfony, por lo que es un sistema multiplataforma soportado tanto por

Windows como por sistemas Unix.

Originalmente fue escrito por Dries Buytaert como un foro de opinión. En el año 2001 se convirtió en un proyecto de código abierto y ganó popularidad en 2004 al utilizarse para crear el sitio 'DeanSpace' para el candidato a las primarias por el Partido Demócrata de Estados Unidos Howard Dean.

Actualmente Drupal se encuentra en la versión 8, aunque aún brinda soporte a la versión 7 que cuenta con muchos adeptos. La mayoría de los módulos y temas de Drupal desarrollados por sus usuarios son compatibles con la versión 7.

Algunos ejemplos de sitios desarrollados con Drupal son las webs de Ubuntu, Intel o Economist.

## Angularjs



**Figura 2.7:** Logo de Angularjs

Angularjs es un framework para aplicaciones web de código abierto basado en Javascript especializado en aplicaciones single-page (SPA). Actualmente está soportado por Google y una comunidad formada por desarrolladores individuales y empresas. Fue inicialmente programado en el año 2009 por Miko Hevery como un software base para una aplicación on-line de almacenamiento de datos en JSON, aunque debido al pobre acogimiento que tuvo la idea, el producto se desechó y se lanzó Angular como una librería de código abierto. Inicialmente todo el contenido de Angularjs iba en un único paquete pero en las versiones más recientes, se ha modularizado para que cada desarrollador tenga una instalación personalizada de acuerdo a sus necesidades.

Actualmente tenemos dos vertientes de Angularjs, la versión 1.x y la versión 2.x. Aunque compartan el nombre, la Angularjs 2.x no se trata de la siguiente iteración del framework, si no que se trata de un remodelado completo incompatible con la versión anterior, por lo que se tratan como si fueran dos librerías diferentes y ambos continúan

recibiendo soporte.

El framework se basa en el modelo vista presentador con lo que nos brinda ficheros de vista (HTML), un modelo de datos y un presentador (controlador). Funciona mediante ciclos de digest, en los cuales Angularjs comprueba todos los scopes (ámbitos) y busca todas las variables que hayan podido sufrir cambios para actualizar las vistas. También al cambiar un valor en una vista se actualiza la variable asociada en el controlador (two-way data binding). Angular está compuesto por módulos y directivas, y los desarrolladores pueden crear sus propios elementos y compartirlos con otros a través de npm o bower. Algunas de las directivas estándar son: ng-app, ng-view, ng-controller ó ng-model.

En este proyecto se usa la versión 1.6 del framework. Algunos sitios desarrollados con Angular son Wolfram Alpha, ABC News ó NBC

## JQuery



**Figura 2.8:** Logo de JQuery

Es una librería de Javascript diseñada para simplificar la programación en el lado del cliente. Es de código abierto y es la librería más usada de Javascript. Fue lanzado en el año 2006 por John Resig quien se inspiró en las librerías de cssQuery.

Principalmente contiene funciones para manipular el DOM, realizar animaciones, gestionar eventos y desarrollar aplicaciones basadas en Ajax. Sus principales características son las siguientes:

- Selección de elementos DOM
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath
- Eventos
- Manipulación de la hoja de estilos CSS

- Efectos y animaciones
- AJAX
- Soporte de extensiones

Ejemplo:

```
$(document).ready(function() {
    console.log('¡Hola Mundo!');
});
```

## JsPDF

JsPDF es una librería Javascript para la generación de PDFs en el lado del cliente. Cuenta con un potente motor para dibujar elementos geométricos, incrustar imágenes y textos variando el tipo de letra y el tamaño. Su primera versión fue lanzada en el año 2012 y desde entonces sus creadores (Parallax) han ido incorporando funcionalidades y librerías para aumentar sus capacidades.

## Dragabilly

Se trata de una librería creada para JQuery que permite utilizar las características de Drag and Drop en una página web. La ventaja que nos ofrece es que podemos mover un elemento web a cualquier punto de un contenedor. Cuenta con un wrapper para Angularjs que nos permite añadirlo como una directiva para ser usado directamente sin necesidad de JQuery. En el proyecto es usado para mover los elementos de una plantilla PDF.

## Html2Canvas

Html2Canvas es una librería Javascript para la conversión de código HTML al objeto Canvas de HTML5. Este objeto permite renderizar la página web con todos sus estilos y guardarla como si fuera una imagen. Al ser compatible con JsPDF nos permite la generación de un documento PDF en modo imagen tal y como se muestra en la web y sin los problemas derivados de los estilos css de impresión.

### 2.3.3. Sistemas

#### Apache HTTP Server



**Figura 2.9:** Logo de Apache HTTP Server

Llamado coloquialmente Apache, es un servidor web multiplataforma de código abierto. La Apache Software Foundation es la encargada actualmente de dar soporte y programación del producto. Su desarrollo comenzó en el año 1995 y jugó un papel importante en el crecimiento de la World Wide Web sustituyendo rápidamente a NCSA HTTPd como servidor HTTP dominante en el mercado. Actualmente es el servidor más utilizado del mundo siendo utilizado por un 46 % de los sitios web en activo.

Entre sus principales características se encuentra el soporte para lenguajes de programación como Perl, Python, Tcl y PHP, módulos de autenticación (`mod_access`, `mod_auth`, `mod_digest` y `mod_auth_digest`), soporte para SSL y TLS, hosting virtual, y autenticación por contraseña o firma digital entre otros.

## MySQL



**Figura 2.10:** Logo de MySQL

MySQL es un sistema gestor de base de datos relacionales de código abierto creado por la compañía sueca MYSQL AB en el año 1995. Durante este tiempo MySQL fue ganando popularidad y en el año 2008 fue comprado por Sun Microsystems para posteriormente pasar a manos de Oracle, actuales propietarios. Antes de la adquisición por parte de Oracle, uno de los creadores y fundadores de MYSQL AB, Monty Widenius, realizó un fork de la versión 5.5 de MySQL para crear MariaDB y mantener la esencia del producto.

Actualmente MySQL se encuentra en la versión 5.7 y se ofrece tanto la versión MySQL Community Server como Enterprise Server, siendo ésta última una licencia de pago. Aun-

que MySQL server ha evolucionado para poder albergar y tratar volúmenes altos de datos, se sigue utilizando principalmente para aplicaciones de pequeño a mediano tamaño.

Sus principales características son:

- Sistema de base de datos relacional
- Arquitectura cliente-servidor
- Compatible con el estándar de SQL
- Sentencias SELECT anidadas
- Vistas
- Procedimientos almacenados y triggers
- Transacciones

#### 2.3.4. Herramientas de desarrollo

Las herramientas de desarrollo nos sirven para facilitarnos la vida a la hora de programar y gestionar recursos. A continuación se describen las principales herramientas usadas en el proyecto.

##### Brackets



**Figura 2.11:** Logo de Brackets

Brackets es un editor de texto avanzado escrito en HTML, CSS y Javascript por Adobe Systems y que está orientado al desarrollo web. A diferencia de otros editores, contiene un pequeño servidor Nodejs para probar el código realizado en el momento. Es capaz de mostrar un correcto formato de múltiples lenguajes de programación como Javascript, PHP, Java, HTML y CSS entre otros. Es un programa de libre distribución y sus desarrolladores han publicado una API con la que se pueden desarrollar módulos para aumentar su funcionalidad. Su interfaz es sencilla, con una lista de archivos abiertos a la

izquierda y espacio para escribir a la derecha ocupando el máximo espacio posible de la ventana. Sus principales características son:

- Edición rápida
- Acceso a archivos de forma rápida
- Previsualización en tiempo real
- JSLint
- Soporte para LESS
- Integración con Theseus
- Extensible

## Git



**Figura 2.12:** Logo de Git

Es un sistema de control de versiones cuya función principal es la de mantener un registro de cambios locales realizados en un directorio y coordinar el trabajo sobre esos archivos entre un grupo de gente. Se usa mayoritariamente como gestión de código a la hora de desarrollar software de manera distribuida. Fue creado por Linus Torvalds en el año 2005 para el desarrollo del kernel de Linux con la finalidad de coordinar el trabajo con otros desarrolladores. A Torvalds no le convencían otros sistemas de control de versiones disponibles. Una de las características de Git es que cada directorio de Git en cada ordenador es un repositorio completo con histórico y rastreo de versiones independiente de cualquier servidor central. Esto hace posible que, en caso de perderse el repositorio remoto, este se pueda rehacer desde uno perteneciente a los desarrolladores. Sus principales características son:

- Soporte para desarrollo no lineal
- Desarrollo distribuido

- Compatibilidad con varios protocolos de comunicaciones, como HTTP, FTP o SSH
- Manejo eficiente de proyectos grandes

## Bower



**Figura 2.13:** Logo de Bower

Bower es una herramienta de gestión de dependencias utilizado principalmente para gestionar componentes HTML, CSS y Javascript y obtener las últimas versiones de los mismos con tan sólo usar un comando de consola. Para instalar Bower, se requiere tener instalado previamente Nodejs, npm y git. La forma más sencilla de instalar un nuevo paquete es buscar en internet el nombre del paquete que deseemos y posteriormente ejecutar en una consola de comandos *bower install nombre\_paquete*. También es posible tener un archivo llamado *bower.json* en el que se guardan todas las dependencias de nuestro proyecto (con nombre de paquete y versión), de esta forma al ejecutar *bower install*, se instalaran todos los paquetes de nuestro fichero, en la versión especificada. Es posible también utilizar sentencias condicionales en las versiones.





## Capítulo 3

# Metodologías de desarrollo software

Una metodología de desarrollo software, es un framework utilizado para estructurar, planificar y controlar el proceso de desarrollo de software. Tanto escogiendo metodologías tradicionales o ligeras, de su correcta implantación en el proyecto dependerá mucho el éxito o fracaso del mismo.

### 3.1. ¿Por qué usar metodologías?

Hoy en día el número de empresas y proyectos que fallan en su ejecución por una mala gestión es muy alto, un gran número de proyectos son ejecutados sin usar una metodología clara. Esto es debido, en numerosas ocasiones, a que el cliente encuentra dificultad a la hora de ver el valor inmediato de los requerimientos del negocio, análisis de casos de uso y especificaciones de diseño que se requieren para generar un producto de calidad. En vez de eso, sienten que pueden simplemente decir lo que desean y que los desarrolladores serán capaces de dárselo sin hacer preguntas adicionales. Sin una metodología clara, los proyectos se suelen entregar tarde, con sobrecostos, y en muchos casos sin alcanzar las expectativas del cliente o del usuario final. Puede incluso llevar al fracaso del proyecto.

Algunas de las ventajas de implantar una metodología bien definida son las siguiente:

- Permite realizar mejores estimaciones
- Entregar sistemas estables
- Mantener al usuario informado
- Preveer futuras tareas
- Gestión del riesgo identificando posibles problemas
- Mejor gestión de tiempo



Cuando una metodología no se implementa correctamente, pueden surgir numerosos problemas. Por ejemplo, una pérdida de comunicación entre el cliente y el equipo de desarrollo lleva a desarrollar sistemas que no cumplen las expectativas del cliente. Además el mal uso de conceptos o procesos de una metodología de desarrollo conduce a menudo a desarrollar sistemas con numerosos defectos.

Por otro lado, la calidad de vida del equipo humano también se ve afectada por la falta de uso de metodologías. Muchos desarrolladores sufren de falta de moral y motivación a causa de los continuos cambios de alcance o parches debidos a la falta de definición en las tareas. Por el contrario, usando los procesos de desarrollo adecuados, se puede gestionar debidamente el alcance de un proyecto y evitar muchos problemas comunes de desarrollo software. Siempre ocurrirán imprevistos, pero con una metodología adecuada, estos son de poco impacto y mínima ocurrencia.

No todas la metodologías de desarrollo funcionan con todos los proyectos. Elegir la adecuada también es uno de los retos que tienen que afrontar los desarrolladores. Algunas metodologías funcionan mejor con proyectos grandes y de larga duración, y otras son más adecuadas para proyectos medianos y pequeños con equipos de desarrollo más modestos. Sin embargo es un hecho que, aplicar una metodología de desarrollo a un proyecto, aumenta mucho las posibilidades de que termine con éxito.

## 3.2. Metodologías tradicionales vs ágiles

Dentro de las metodologías de desarrollo software, podemos encontrar una división entre metodologías tradicionales (o metodologías pesadas) y metodologías ágiles.

Las metodologías de desarrollo tradicionales se caracterizan por tener una serie secuencial de fases: requisitos, análisis, diseño, implementación, pruebas y despliegue. Primero, los requisitos del cliente son meticulosamente documentados. Después se analiza y diseña la arquitectura general del software y en la siguiente fase comienza la implementación, finalizando con el período de pruebas y el despliegue. La idea principal es tener una visión completa del proyecto antes de su inicio y realizar la implementación de acuerdo al esquema planificado.

- Desarrollo Predictivo

- Las decisiones se toman en las primeras fases del ciclo de vida y se especifican mediante modelos gráficos
- Las decisiones se interpretan y materializan en la fase de desarrollo
- Orientado a la documentación
- Orientado al proceso

Las metodologías ágiles son menos rigurosas a la hora de gestionar la documentación. La clave reside en que el desarrollo se realiza de manera incremental e iterativa, en la que las fases de diseño se ejecutan una y otra vez. Como resultado, el software deja de ser tan rígido para convertirse en una estructura más orgánica con mayor interactividad entre sus componentes. Se le da mayor importancia a la adaptabilidad y a las pruebas de integración.

- Preparado para soportar cambios
- Desarrollo iterativo
- Pruebas de integración
- Orientado al código
- Orientado a las personas

TRADICIONALES	ÁGILES
Proceso secuencial	Proceso iterativo e incremental. Fases en paralelo
Estructurado	Flexible. Adaptable. Mejora continua
Grandes y pocas entregas de software	Entregas pequeñas y frecuentes de software, que siempre aportan valor
Un gran proyecto	Muchos pequeños proyectos
Adecuado para situaciones donde los cambios no son frecuentes	Adecuado cuando el usuario no conoce al detalle todas las necesidades al principio, y, además, serán cambiantes
El usuario participa fundamentalmente en la fase inicial de toma de requisitos	El usuario está involucrado constantemente en el proyecto. Comunicación fluida
Buena definición de requisitos al principio	Los requisitos se van desgranando cuando se van a abordar
Plazos y costes son conocidos al principio	Ámbito y costes varían según evoluciona el proyecto
Proyecto bien documentado	La documentación no es prioritaria



## Capítulo 4

### Descripción Informática (20-35 %)

Para ello describirás:

**4.1. La base de datos coleccionada (si tiene sentido).**

...

**4.2. Los algoritmos para el desarrollo de la solución**

...



## Capítulo 5

### Conclusiones y trabajos futuros (5 %)

Resumen de los logros principales conseguidos, destacando:

#### **5.1. Implementación**

...

#### **5.2. Resultados**

...

En futuros trabajos, a partir de una crítica constructiva del trabajo realizado, plantear mejoras y extensiones del mismo.





# Bibliografía

- [NC97] Isaac Newton and Naomi Campbell. A re-formulation of gravity with respect to really cool models. *Jornal of Funny Physics*, 35:39–78, 1997.
- [Pes03] Mark D. Pesce. *Programming Microsoft DirectShow for digital video and television*. Microsoft Press, 2003.



# Anexo 1

Lo relativo al anexo 1 ...

Ejemplos de tratamiento de texto:

Probamos una cita [NC97]

Citamos un libro[Pes03]

Probamos a poner una nota al pié<sup>1</sup>

Probando: *cursiva* **negrita** subrayada enfatizar

---

<sup>1</sup>Mi primera nota al pié