

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

Grado en Ingeniería Informática

Trabajo Fin de Grado

**Implementar un e.commerce con software libre para plataformas
móviles**

Autor: Mario Gutiérrez Calvo de Mora

Director: José Luís Castillo Sequera

TRIBUNAL:

Presidente:

Vocal 1:

Vocal 2:

CALIFICACIÓN:

FECHA:

Agradecimientos

A mis padres y mi hermana, que tanto me han apoyado y comprendido, aportándome ánimos en los buenos y malos momentos.

A mis amigos, que han sabido apreciar y entender el esfuerzo que he empleado para llegar hasta aquí.

A José Luís Castillo, por aportarme la base y despertar mi interés por esto, además de la ayuda prestada durante el desarrollo.

1.1 Resumen

En la actualidad, el comercio electrónico está a la orden del día, aumentando exponencialmente el número de usuarios que utilizan sus dispositivos móviles para realizar sus compras.

Gracias a plataformas como Webkit, se pueden desarrollar Aplicaciones Web que funcionen en prácticamente cualquier dispositivo, ya sea un ordenador o un dispositivo móvil.

El objetivo de este TFG es profundizar, analizar y crear mediante una herramienta como CMS de Magento, una tienda online, que además esté orientada a los usuarios que usan dispositivos móviles, usando la interconexión de lenguajes como son: php, JavaScript, CSS, HTML...sobre la plataforma Webkit.

1.2 Abstract

At present, e-commerce is the order of the day; it's exponentially increasing the number of users that using their mobile devices for shopping.

Thanks to platforms like Webkit, you can develop Web applications that run on virtually any device in a computer or a mobile device.

The aim of this TFG is to deeper, to analyze and to create with a tool such as CMS Magento, an online store, which also is aimed at users who use mobile devices, using the lenguajes interconnection such as: PHP, JavaScript, CSS, HTML...on the Webkit platform.

1.2.1 Palabras clave/Keywords

WebKit, Magento, CMS, e-commerce, Mobile

Índice General

1.1	Resumen	5
1.2	Abstract	5
1.2.1	Palabras clave/Keywords	5
Capítulo 1: Introducción		11
1.3	Definiciones, acrónimos y abreviaturas	11
1.4	Objetivos generales	11
1.5	Finalidad	12
1.6	Alcance	12
1.7	Hipótesis	12
1.8	Plan de negocio	13
1.8.1	Categoría E-Commerce Elegida	13
1.8.2	Características, Operatividad y Propiedades de Ópticas Zaera	13
1.8.3	Tipo de Productos, Categorías y Precios	13
1.8.4	Forma de Pedido, Entrega y Pago	16
1.8.5	Publicación, Promoción y Anuncio	18
1.8.6	Modelo de negocio	23
1.8.7	Plan de Proyecto. Diagrama de Costes	25
2	Capítulo 2: Desarrollo/implementación	37
2.1	File system Magento	37
3	Desarrollo/implementación basado en WebKit	37
3.1	WebKit: Indiferencia entre iOS y Android	38
3.1.1	WebKit y HTML5	38
3.2	Consideraciones sobre la aplicación web Móvil	39
3.2.1	Presentación Genérica del sitio	39
3.2.2	Mobile-friendly	39
3.2.3	Hecho para móviles	43
3.2.4	La construcción de la aplicación	48
3.3	Crear una aplicación basada en el navegador siendo multiplataforma	54
<i>El uso de Ajax, jQuery, JavaScript, CSS Y HTML5</i>		54
3.3.1	Introducción	54
3.3.2	La aplicación	55
3.3.3	Consideraciones de diseño	55
3.3.4	La construcción de la aplicación	56
3.3.5	El lado del cliente – La vista por defecto	58
3.3.6	El lado del cliente – El HTML	59

3.3.7	El lado del cliente – Rellenar la lista	60
3.3.8	El lado del cliente – Actualizar con Ajax	65
3.3.9	El lado del servidor – Construir tu propia página para monitorear	66
3.4	Magento mobile.....	69
3.5	Desarrollo/Implementación de la Aplicación web para la tienda	73
3.5.1	FILE SYSTEM.....	73
3.5.2	Excepción para funcionamiento de la Aplicación web:.....	76
3.5.3	Activación del módulo Desarrollador Magento.....	78
3.5.4	Funcionalidades	80
3.5.5	Formato y diseño	82
4	Pruebas	87
4.1	Pruebas referentes a la aplicación web desarrollada para la tienda	87
4.1.1	Pruebas unitarias	87
4.1.2	Pruebas de usabilidad	88
5	Conclusiones	96
1	Referencias Bibliográficas	98
2	Anexos	100
2.1	Manual desarrollo Magento	100
2.2	Manual usuario aplicación web.....	100
2.3	Creación imagen superior	100

ÍNDICE DE FIGURAS

Figura 1. Análisis DAFO 1.....	19
Figura 2. Análisis DAFO 2.....	19
Figura 3. Análisis DAFO 3.....	19
Figura 4. Tres Escenarios futuros.	22
Figura 5. Costos del proyecto.....	26
Figura 6. Costos por recurso:.....	26
Figura 7. Costos por asignación:.....	27
Figura 8. Diagrama de Gant.....	30
Figura 9. Diagrama de Gant desglosado:.....	30
Figura 10. Vista Original tienda.	37
Figura 11. Ejemplo web original adaptada al dispositivo.....	41
Figura 13. Configuración navegador Android.....	42
Figura 14. Diseño explorador escritorio.....	43
Figura 15. Diseño explorador móvil.	44
Figura 16. Diseño explorador móvil 2.....	44
Figura 17. Diseño explorador móvil 3.....	45
Figura 18. Versión móvil.....	45
Figura 19. Enlace versión original.....	46
Figura 20. Estadísticas usuarios móviles.	47
Figura 21. Aplicación vista desde el navegador Safari en el escritorio.	50
Figura 22. Aplicación en el navegador de Android.....	50
Figura 23. Aplicación en el navegador de iPhone.	50
Figura 24. Captura de la estructura de la base de datos.....	58
Figura 25. Los sitios que no funcionan del todo bien.....	58
Figura 26. Forma simple.....	63
Figura 27. Herramienta de validación de JSON.....	68
Figura 28. Aplicación que se ejecuta en iPhone.....	68
Figura 29. Aplicación que se ejecuta en Android.....	68
Figura 30. Instalación Extensión Magento Mobile.....	69
Figura 31. Instalación Extensión Magento Mobile 2.....	70
Figura 32. Instalación Extensión Magento Mobile 3.....	70
Figura 33. Instalación Extensión Magento Mobile 4.....	70
Figura 34. Instalación Extensión Magento Mobile 5.....	71

Figura 35. Instalación Extensión Magento Mobile 6.....	71
Figura 36. Instalación Extensión Magento Mobile 7.....	71
Figura 37. Instalación Extensión Magento Mobile 8.....	72
Figura 38. Estructura de carpetas general de Magento.....	73
Figura 39. Estructura de carpetas diseño app.....	75
Figura 40. Estructura de carpetas diseño app (template).....	75
Figura 41. Archivos phtml principales de la app.	76
Figura 42. Estructura de carpetas diseño app (skin).	76
Figura 43. Creación de la excepción.....	77
Figura 44. Sistema/Configuración.	78
Figura 45. Módulo desarrollador de Magento.....	78
Figura 46. Módulo desarrollador de Magento 2.	79
Figura 47. Cambio de vista para aplicar las propiedades.....	79
Figura 48. Activación del módulo desarrollador de Magento.....	79
Figura 49. Referencia de los diferentes contenedores de la Web.	80
Figura 50. Creación página app desde el backend.....	81
Figura 51. Creación de enlace a la app en el código.	81
Figura 52. Muestra del enlace.....	82
Figura 53. Modificación en el código para el logo.....	83
Figura 54. Modificación en el código para el logo 2.....	83
Figura 55. Cambio estilo app.....	83
Figura 56. Cambio estilo app 2.....	83
Figura 57. Página análisis de estilos.	84
Figura 58. Resultado análisis de estilos.....	84
Figura 59. Extensión para edición de formatos y estilos.....	85
Figura 60. Generador de código CSS.	86
Figura 61. Cambio del color general de la aplicación.....	86
Figura 62. Se aplica el color al menú de categorías.	87
Figura 63. Prueba app en iOS 7.	87
Figura 64. Prueba app en Windows Phone 8.0.	87
Figura 65. Prueba de usabilidad para la vista original de la tienda.....	88
Figura 66. Resultados de la prueba de usabilidad para la vista original de la tienda. ...	89
Figura 67. Resultados de la prueba de usabilidad para la vista original de la tienda 2..	89
Figura 68. Lista de las imágenes que contiene la página.	91

Figura 69. Optimización de imágenes.	91
Figura 70. Prueba de usabilidad vista de la aplicación.....	93
Figura 71. Resultado de la prueba de usabilidad vista de la aplicación.	94
Figura 72. Resultado de la prueba de usabilidad vista de la aplicación 2.	95
Figura 73. Listado de imágenes que carga la vista de la aplicación.	96

Capítulo 1: Introducción

1.3 Definiciones, acrónimos y abreviaturas

CMS (Content Management System): Sistemas gestor de contenidos, que permite crear un framework para crear y administrar contenidos. Está enfocado principalmente a páginas web.

M-commerce: tipo de comercio que toma sus bases en el e-commerce, solo que todas las transacciones se hacen desde un dispositivo móvil.

Web accesible: debe ser utilizable para el mayor número de usuarios, sin tener en cuenta sus conocimientos o sus capacidades personales, además de intentar abarcar el mayor número de dispositivos que intenten acceder a la web.

Movilidad: tener accesible la información, independientemente del lugar.

Escalabilidad: el sistema tendrá la propiedad y capacidad para adaptarse sin perder calidad y también tener un crecimiento continuo de manera fluida.

SaaS: los datos y el soporte lógico lo maneja una empresa TIC.

Estilo Hover: cuando un objeto de la interfaz gráfica se ve alterado al pasar el puntero sobre él, sin llegar a seleccionarlo.

JSON (JavaScript Object Notation): formato para intercambio de datos, surge como alternativa a XML, más compatible con JavaScript. La ventaja es que puede ser leído por cualquier lenguaje de programación. Usado para el intercambio de información entre distintas tecnologías.

1.4 Objetivos generales

El objetivo de este Trabajo de Fin de Grado, es poder desarrollar un e-commerce gracias a una potente plataforma OpenSource como es MAGENTO. Ampliar el aprendizaje personal en la materia, así como una gran habilidad para emplear un gestor de contenidos.

Como primer objetivo será la implementación de una tienda configurando todos los módulos que incorpora de serie, dotando el e-commerce de total funcionalidad, usabilidad y accesibilidad; poder integrarlo con redes sociales, así como recoger y analizar infinidad de datos integrándolo con Google Analytics.

Por último, el objetivo principal sería la idea de llegar a conseguir abarcar más mercado, abriendo horizontes hacia los usuarios que usan dispositivos móviles para realizar sus compras. Este nicho de mercado es cada vez más amplio. Para ello se realizará una aplicación web basada en la plataforma Webkit, que no distinguirá entre el sistema operativo desde el dispositivo móvil que se accede, ya que trabaja sobre el navegador del mismo. Adaptado más a las necesidades de este tipo de usuarios

1.5 Finalidad

El objetivo final, para el que está enfocado este proyecto, es comprobar y hacer posible que la tienda implementada sea totalmente funcional y visualmente atractiva, y además esté adaptada a cualquier dispositivo móvil mediante la plataforma Webkit. Por lo cual tendremos una versión original y una versión móvil, en la cual habrá que simplificar información menos relevante, para que así sea más clara y concisa para este tipo de dispositivos.

1.6 Alcance

Para identificar el alcance existen varios tipos de información:

- Entregables dentro y fuera del alcance, como pueden ser los requerimientos de negocio y el análisis de la situación actual.
- El ciclo de vida, los procesos de éste tanto fuera como dentro del alcance, por ejemplo: análisis, diseño, pruebas, etc.
- Los tipos de datos de fuera y dentro del alcance (datos financieros, de ventas...).
- Las bases de datos de fuera y dentro del alcance como son las de facturación, nóminas, activos, etc.
- Las organizaciones, tanto las de dentro como las fuera del alcance, como por ejemplo las de recursos humanos o las de proveedores.
- Las funcionalidades de dentro y fuera del alcance, soporte de decisiones, capturas de datos, reportes de gestión...

1.7 Hipótesis

WebKit es una plataforma para aplicaciones que funciona para gran variedad de navegadores, los cuales son:

Konqueror (desde su versión 4.5 tiene la opción para usarlo, desde la 4.6 tiene la opción para usarlo por defecto), Safari, Google Chrome (antes de usar Blink), Chromium (antes de usar Blink), Epiphany, Midori, QupZilla, Arora, Rekonq, Swift, OmniWeb (desde su versión 5.5), Shiira, Web Browser for S60, SRWare Iron (bifurcación de Google Chrome), Palm Pre & Pixi, RockMelt, Maxthon, Playstation 3 (navegador) (desde su versión de firmware 4.10), Mijitt Explorer (Navegador de Mijitt OS), Opera (desde la versión 15), WebPositive.

He elegido este tipo de plataforma ya que de tal forma se afina mejor al cliente, siendo más atractiva visualmente y no liberando la página de funciones menos importantes.

Dicha plataforma permite trabajar de forma transparente en prácticamente cualquier sistema operativo, debido a su alta funcionabilidad e integración. Por ello esto aporta a mi proyecto un punto de vista enfocado más a la usabilidad y eficacia, pues para llegar a un cierto resultado no son necesarias muchas herramientas, es decir, no es necesario

utilizar una plataforma distinta para cada sistema operativo a la hora de desarrollar la aplicación móvil. Además existe la gran ventaja de que WebKit es Open Source.

1.8 Plan de negocio

1.8.1 Categoría E-Commerce Elegida

Se ha optado por elegir la categoría B2C, ya que se dispone de una tienda física, y se quiere ampliar las fronteras de venta, con la creación de un modelo de E-Commerce B2C, para ofrecer los productos y llegar al mayor número de clientes o futuros potenciales clientes.

1.8.2 Características, Operatividad y Propiedades de Ópticas Zaera

La empresa, trata de una óptica, con varias tiendas en la provincia de Galicia. Lo que se quiere, es ampliar fronteras más allá de la propia comunidad, para llegar a más clientes, y poder así realizar el mayor número de ventas posibles, al tratarse de una óptica, para ventas fuera de la comunidad.

La empresa se denomina “Ópticas Zaera”.

Contamos con tres establecimientos en los que opera en:

- A Coruña
- Vigo
- Bertamiráns (A Coruña)

Con el sitio web, se tratará de operar a lo largo de todo el territorio nacional con la venta de productos, se pretende fomentar sobre todo la venta de gafas de sol, productos de limpieza, y diversos productos relacionados (líquidos de lentillas, líquidos para la limpieza ocular, fundas...) con la sanidad ocular.

Además si se nos facilita, previa visita al oftalmólogo, la graduación de cristales, también se realizará la venta de gafas graduadas, con los datos del oftalmólogo.

Además en los pedidos de gafas, previo al pedido, se pedirán unos datos imprescindibles, para que las monturas de las gafas, se adapte, perfectamente a la cara de los clientes, como son:

- Longitud de las lentes
- Longitud del puente
- Ancho total de la montura.
- En caso de las gafas sean graduadas, o se trate de unas lentillas, se pedirán los datos facilitados por el oftalmólogo

1.8.3 Tipo de Productos, Categorías y Precios

En la tienda virtual vamos a ofrecer todos los productos relacionados con el sector de la óptica.

En dicha página web se ofrecen tanto gafas de sol, como gafas graduadas, lentes de contacto y limpiadores de dichas lentes.

Se han subdividido los productos en distintas categorías, y éstas a su vez, en distintas subcategorías, según la marca del producto, como se puede observar, estos son algunos ejemplos:

Tabla 1. Productos de limpieza:

MARCA	MODELO	PRECIO	AÑO	CODIGO REFERENCIA	Cantidad	Porta lentillas	Existencias
SYSTANE	SYSTANE ULTRA 10 ml	10	2014	SYST_10ML	10 ml	No	1000
RENU	ReNu Multi-Purpose Solution 360ml	5,9	2014	REN_MP_360	360 ml	Si	1000
RENU	ReNu MultiPlus 360ml	9,9	2014	REN_MLP_360	360 ml	Si	1000
RENU	ReNu MultiPlus 6x240ml	36	2014	REN_MLP_1440	1440 ml	Si	1000
AOSEPT	AOSEPT Plus 360ml	15,9	2014	AOS_360	360 ml	No	1000
AOSEPT	TriPack Aosept Plus 3x360 ml +90ml	42,5	2014	AOS_3X360	1170 ml	Si	1000

Tabla 2. Lentes de Contacto (Lentillas):

MARCA	MODELO	PRECIO	Tipo	Duración	AÑO	CODIGO REFERENCIA	Existencias
ACUVUE	1 Day Acuvue	26	Blancas	Mensual	2014	AC_1D	100
ACUVUE	1 Day Acuvue Moist 90 Pack	60	Blancas	Mensual	2014	AC_1D_M_90	100
ACUVUE	1 Day Acuvue Moist for Astigmatism	38	Blancas	Mensual	2014	AC_1D_M_A	100
ACUVUE	Acuvue 1-Day TruEye 90 Pack	35	Blancas	Mensual	2014	AC_Acuvue 1A_T_P_90	100
ACUVUE	Acuvue 2	22	Duras	Anual	2014	AC_Acuvue 2A	100
ACUVUE	Acuvue 2 Colors Opaques	37	Blancas	Mensual	2014	AC_Acuvue 2A_O	100
BIOMEDICS	Biomedics 38	16	Duras	Anual	2014	BIO_38	100
BIOMEDICS	Biomedics 55	16	Duras	Anual	2014	BIO_55	100
BIOMEDICS	Biomedics 55 Evolution	22	Blancas	Mensual	2014	BIO_55_EVO	100
BIOMEDICS	Biomedics Toric (UltraFlex Toric)	32	Blancas	Mensual	2014	BIO_TOR	100
BIOMEDICS	Biomedics XC	28	Blancas	Mensual	2014	BIO_XC	100
VERTEX	Vertex Sphere (Encore Sphere)	30	Duras	Anual	2014	VX_S	100
VERTEX	Vertex Toric (Encore Toric)	33	Blancas	Anual	2014	VX_T	0
VERTEX	Vertex Toric XR (Encore Toric XR)	81	Blancas	Mensual	2014	VX_T_XR	100

Tabla 3. Monturas para Gafas graduadas (Precio de la montura sin el cristal):

MARCA	MODELO	PRECIO	GENERO	TAMAÑO	PESO (gr)	MATERIAL	Estuche	COLOR	AÑO	CODIGO REFERENCIA	Existencias
ARNETTE	Arnette AN6001	66,95	H-M	S/M/XL	25	Metal	Si	Negro	2014	AN6001	100
ARNETTE	Arnette AN6007	66,95	H-M	S/M/XL	35	Metal	Si	Azul	2014	AN6007	100
ARNETTE	Arnette AN6016	66,95	H-M	M	32	Metal	Si	Verde	2014	AN6016	15
ARNETTE	Arnette AN6018	68,95	H-M	M	30	Metal	Si	Negro	2014	AN6018	15
ARNETTE	Arnette AN6020	69,95	H-M	M	35	Metal	Si	Negro	2014	AN6020	15
BLUE BAY	Bluebay B&B 628	66,95	H	M	31	Aluminio	Si	Negro	2014	B&B 628	15
BLUE BAY	Bluebay B&B 676	59,95	M	M	24	Aluminio	Si	Negro	2014	B&B 676	15
BLUE BAY	Bluebay B&B 683	52,95	H	M	28	Aluminio	Si	Negro	2014	B&B 683	15
BLUE BAY	Bluebay B&B 684	52,95	M	S/M/XL	29	Aluminio	Si	Negro	2014	B&B 684	15
BLUE BAY	Bluebay B&B 685	55,95	M	S/M/XL	34	Aluminio	Si	Marrón	2014	B&B 685	15
BURBERRY	Burberry BE1005	135,65	H-M	S/M/XL	24	Aluminio	Si	Amarillo	2009	BE1005	15
BURBERRY	Burberry BE1007	136,95	H-M	S/M/XL	18	Aluminio	Si	Negro	2010	BE1007	15
BURBERRY	Burberry BE1009	135,85	H-M	S/M/XL	20	Aluminio	Si	Negro	2011	BE1009	15
BURBERRY	Burberry BE1010	140	H-M	S/M/XL	21	Aluminio	No	Negro	2012	BE1010	15
BURBERRY	Burberry BE1014	170	M	S/M/XL	25	Aluminio	Si	Negro	2014	BE1014	15
BVLGARI	Bvlgari BV1002T	241,95	H-M	S/M	35	Aluminio	Si	Azul	2014	BV1002T	15

Tabla 4. Gafas de Sol:

MARCA	MODELO	PRECIO	GENERO	TAMAÑO	PESO (gr)	CRISTAL	COLOR	MATERIAL	Estuche	AÑO	CODIGO REFERENCIA	Existencias
ARNETTE	Arnette AN127	57,95	M	S/M/XL	20	Oscurecido/Negro/Marrón	Negro	Aluminio	Si	2008	AR_AN127	15
ARNETTE	Arnette AN3001	66,95	H	S/M/XL	21	Negro	Negro	Aluminio	No	2009	AR_AN3001	15
ARNETTE	Arnette AN3009	69,95	H	S/M/XL	25	Negro	Negro	Aluminio	Si	2009	AR_AN3009	15
ARNETTE	Arnette AN3012	75,95	M	S/M	35	Negro	Negro	Aluminio	Si	2010	AR_AN3012	15
ARNETTE	Arnette AN3021	75,95	H	S/M	32	Marrón/Negro	Azul	Aluminio	Si	2011	AR_AN3021	15
ARNETTE	Arnette AN3025	58,95	H	S/M	30	Negro	Verde	Aluminio	Si	2012	AR_AN3025	15
ARNETTE	Arnette AN3026	58,95	H	S/M	35	Marrón	Negro	Aluminio	Si	2011	AR_AN3026	15
ARNETTE	Arnette AN3048	101,95	H	S/M	31	Oscurecido/Marrón	Negro	Aluminio	Si	2010	AR_AN3048	15
ARNETTE	Arnette AN4007	92,95	H-M	M	24	Negro/Marrón	Negro	Plástico	Si	2009	AR_AN4007	15

ARNETT E	Arnette AN4077P OL Rage XL	85,95	H-M	S/M/XL	28	Negro/ Marrón	Negro	Plástico	Si	2008	AR_AN407 7_XL	15
ARNETT E	Arnette AN4119	93,95	H-M	S/M	29	Negro	Negro	Plástico	Si	2009	AR_AN411 9	15
ARNETT E	Arnette AN4127	89,95	H-M	M	30	Marrón /Negro	Negro	Plástico	Si	2010	AR_AN412 7	15
ARNETT E	Arnette AN4127 Caller	98,95	H-M	S/M/XL	34	Oscure cido/N egro/M arrón	Marrón	Plástico	Si	2008	AR_AN412 7_Caller	15

***Se adjunta documento Excel, con el catálogo y documentos clasificados con las imágenes de los productos.**

1.8.4 Forma de Pedido, Entrega y Pago

A continuación, se indica la forma de pedido, entrega y pago de los productos. (El pago a través del sitio web, se realizará únicamente, a través de Paypal, debido a las comisiones, que conlleva el contratar una pasarela de pago, a través de VISA o Mastercard).

1.8.4.1 Pedido

Se dispone de un sistema completo de gestión de pedidos y facturación. En el que una vez la compra ha sido realizada por el cliente, el panel de administración de la tienda se actualiza con los nuevos pedidos.

Proceso del pedido

Tan pronto como el cliente realiza un pedido, de forma automática se le envía un email con la confirmación a su pedido, junto con un resumen de los datos de este. Al administrador de la tienda, le debería de llegar un email similar.

Después, el cliente recibirá un email confirmando (o no) el pago, indicándole que el pedido está en marcha.

El cliente dispondrá de una intranet, para facilitarle el seguimiento del pedido, donde podrá consultar e imprimir los mismos. Internamente, en el tablero de administración veremos todos los pedidos listados, con sus referencias, cliente el cual ordena el pedido, fecha y estado del mismo.

Para acceder al estado de los pedidos "Ventas/Pedidos". En cada paso del proceso, desde el sistema podremos enviar un email informativo del estado actual en que se encuentra el pedido:

- Procesando
- Pendiente
- En suspenso
- Cancelado
- Cerrado
- Completo

El sistema facilitará una ficha de seguimiento para el producto: donde podemos cambiar el estado del pedido (uso interno) y además notificar al cliente de los cambios que surjan si se cree conveniente, todo ello mediante un formulario. Si el cliente a la hora de la compra del producto a través de tarjeta, dicha cuenta, no tiene fondos, o los fondos son insuficientes, el pedido se quedará en estado "Pendiente".

Los datos a tener en cuenta serán valores por defecto de logística: albarán, factura, etc.

A la hora de realizar un pedido, primeramente se nos pedirán tres datos importantes:

- Método de envío
- Dirección de facturación.
- Método de pago

Cuando ya han sido rellenados estos tres datos, se podrá revisar el pedido.

1.8.4.2 Entrega

Respecto a la gestión de envíos, podemos verlos en Ventas/Envíos, muy fácil y al margen de los demás pedidos. Las funcionalidades son parecidas a la pantalla de pedidos, realizar un listado filtrable, en el que podemos hacer clic en los pedidos para entrar en más detalle.

Desde aquí podremos enviar a los clientes información de seguimiento, y además adjuntar comentarios al pedido.

Las diferentes formas de entrega estarán detalladas de tal forma que el usuario pueda acceder a diferente tipo de información dependiendo de varios factores.

El tiempo de entrega, en el que el cliente podrá comprobar el margen de tiempo en el que le será entregado el producto, dependiendo de:

- La modalidad de transporte que elija. Por ejemplo de forma "Express", en el que el envío será más caro, pero el paquete lo recibirá en el margen de tiempo más pequeño posible.
- El lugar a donde el paquete tenga que llegar (siempre dentro de unas restricciones en las que el paquete no será posible entregar).
- El volumen del paquete, según el peso del paquete a enviar, tendrá una tarifa u otra.

Existirá una tarifa plana de envío para algunos productos, en el que habrá una cantidad fija predeterminada.

Por otro lado, existirán códigos promocionales en los que el cliente a la hora de la compra tendrá la opción de insertarlos mediante un campo, y si el código es validado el envío del producto a comprar será gratuito.

Condiciones de entrega

Ópticas Zaera realizará la entrega a la dirección que cliente facilite, en principio dentro del territorio español. No se realizarán entregas fuera del país.

Siempre que sea posible OpticasZaera realizará entregas parciales, para que el cliente reciba los artículos solicitados lo antes posible.

Los periodos de entrega son indicativos y, por lo mismo, no se consideran fechas límite estrictas. Por el mero hecho de haber de haber excedido un periodo de entrega no dará ningún derecho de compensación al cliente. Para ello deberá enviarse una notificación de incumplimiento.

1.8.4.3 Pago

Los diferentes métodos de pago que ofrece OpticasZaera son:

- Tarjeta de crédito/debito.
- A través de PayPal.

Las tarjetas de crédito/debito que se podrán emplear serán:

- American Express.
- Visa.
- MasterCard.
- Discover.

Para que el cliente realice el pago mediante tarjeta de crédito o débito se le pedirá que introduzca los datos de la tarjeta durante la confirmación de su pedido.

Condiciones sobre métodos de pago

Todas las compras y pedidos en OpticasZaera, podrán ser pagados en principio con las tarjetas de crédito antes descritas y a través de PayPal. OpticasZaera llevará a cabo un control individual de crédito para cada compra. Dependiendo de los resultados de este control, OpticasZaera se reserva el derecho a rechazar ciertos métodos de pago.

En OpticasZaera no se admiten otros medios de pago. Por lo tanto, el cliente no podrá realizar sus pagos en efectivo ni mediante cheques, pagarés, domiciliaciones o transferencias bancarias, contra reembolso o dinero electrónico.

1.8.5 Publicación, Promoción y Anuncio

1.8.5.1 DAFO (Debilidades Amenazas Fortalezas y Oportunidades)

Para comprender un poco la situación de nuestra empresa hemos decidido realizar un estudio que analice nuestras debilidades, amenazas, fortalezas y oportunidades. De esta manera podremos comprender que tipos de promociones, que tipo de marketing y que herramientas de publicidad podemos utilizar.

Para la realización de nuestro análisis, hemos recurrido a una herramienta que te somete a un test y te muestra los resultados de manera gratuita, a través de la siguiente página web:

Figura 1. Análisis DAFO 1.

<http://www.inghenia.com/gadgets/swot/swot.php>

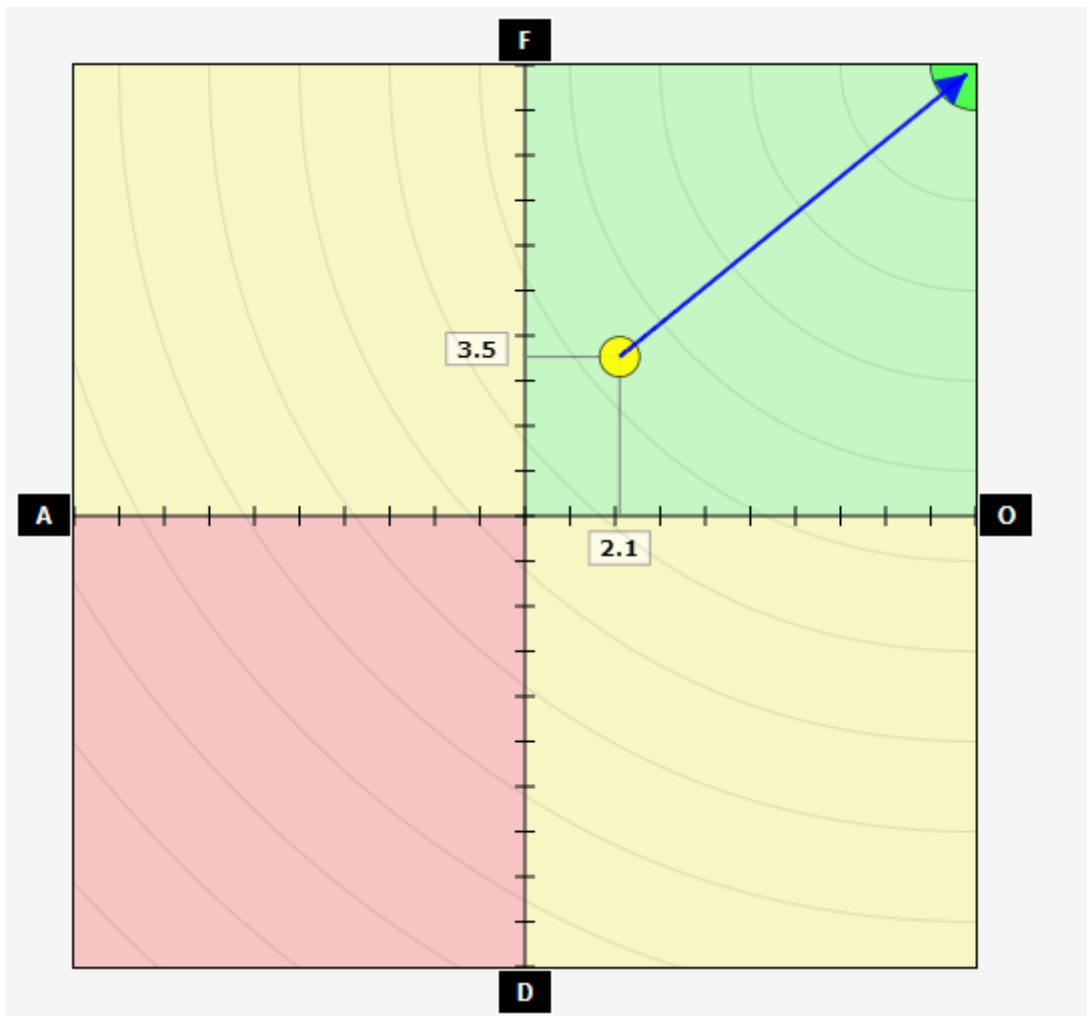
Fortalezas		Peso	Debilidades		Peso
Liderazgo en un Nicho de Mercado		3	Costos		5
Precio Competitivo		9	Innovación de Productos/Servicios		3
Marcas		3	Ubicación Geográfica		2
Alianzas de Negocios		6			
Estado Financiero		7			
Reputación		7			
Presencia Online		8			
Relación con los Proveedores		8			
Personal Experimentado		6			
Cadena de Distribución		4			
Liderazgo en el Mercado		3			
Marketing Efectivo		2			
Originalidad de Productos/Servicios		2			
Plantas y Equipamiento		2			
Total		70	Total		10

Figura 2. Análisis DAFO 2.

Oportunidades		Peso	Amenazas		Peso
*		6	Similaridad de paginas		6
nuevos productos		6	Sencillez de negocio		5
mercado nacional		6	expansion internacional		4
desarrollo de la infraestructura web		6			
Fidelizacion proveedores		6			
Adquisición de Empresa Proveedora de Tecnología		3			
Clientes		3			
Total		36	Total		15

Con lo que obtenemos que nuestra empresa avanza en buen camino, según lo reflejado en los datos anteriores en el siguiente gráfico:

Figura 3. Análisis DAFO 3.



1.8.5.2 Promociones desarrolladas en Magento

- **Reglas de precios en el carrito de la compra:** Reglas impuestas a cualquier tipo de pedido, según la cuantía del pedido del cliente, se hará un descuento en el pedido total:
 - **Para pedidos superiores a 200€:** Se hará un descuento de un total de 10€, en la cuantía total de la compra del cliente.
 - **Para pedidos superiores a 400€:** Se hará un descuento de un total de 30€, en la cuantía total de la compra del cliente.

Con estas reglas, lo que pretendemos, es que los clientes, utilicen nuestra tienda virtual, para realizar la compra completa que haría en uno o dos meses en otra óptica, para que se puedan beneficiar del descuento, o para atraer a otros clientes potenciales, como amigos o familiares, que se podrán aprovechar de este descuento, haciendo un pedido en conjunto.

- **Reglas de precio en el los productos del catálogo:** Reglas impuestas a cualquier tipo de producto, donde se hará un descuento en todos los productos:

- **Días SIN IVA:** Entre los días 19 y 20 de Marzo, pretenderemos acceder a nuevos y potenciales clientes a través de la campaña día sin IVA, en el cual se descontará de los productos un 18 de su valor. Con esto lo que pretendemos será captar nuevos clientes. En caso de que la campaña obtenga éxito, se promoverá otros días a lo largo del año.

1.8.5.3 Publicación de nuestra empresa

Inicialmente vamos a estar presente en todas las redes sociales con el fin de ser más cercanos a la gente y poder realizar campañas de promoción y que su divulgación sea eficiente, rápida y de manera gratuita. Las principales redes sociales en las que estaríamos serían Facebook, Twitter, Tuenti y Google+.

Según vaya creciendo la empresa utilizaremos diferentes métodos de posicionamiento en buscadores (SEM), como por ejemplo sería Google adwords, una herramienta que consiste en los siguientes puntos:

- Los anuncios aparecen simultáneamente en los resultados de búsqueda que muestra el buscador Google.
- Los anuncios aparecen también en zonas de páginas web, en forma de banners, que pueden ser imágenes, vídeos o texto.

También tenemos pensado que nuestra página sea la que se muestre de los primeros resultados cuando un cliente realice una búsqueda con tiendas virtuales relacionadas con los nuestros productos, técnica SEO que también se puede realizar mediante la herramienta de Google Adwords.

Para el pago por estos servicios Google Adwords propone el pago por clic (PPC/ CPC), que consiste en que nosotros como empresa pagamos a Google por cada clic que un usuario realiza mediante eso anuncios.

1.8.5.4 Marketing

Mediante esos métodos de publicación que hemos comentado nuestra campaña de marketing se va a centrar en varias partes, atendiendo al tipo de producto a vender, lo segmentaremos por la edad.

Recordad que nuestros productos ofertados corresponden a gafas de sol, gafas graduadas, lentes de contacto y productos de limpieza.

El producto de gafas de sol lo tendremos destinado a la gente joven, ya que es un sector que implica grandes beneficios, en especial al género femenino.

En cuanto a las gafas graduadas hemos decidido que realizaremos campañas para ofrecer dichos productos a personas de avanzada edad ya que son las que más compras realizan sobre este producto.

Por último tenemos los productos de lentes y limpieza que los ofreceremos también al sector juvenil debido a su gran éxito.

Gracias a la herramienta de Google Analytics que hemos incorporado en nuestra tienda virtual podemos observar datos de compradores mediante su situación geográfica lo que en un futuro puede que derive en que nuestra empresa realice un marketing demográfico.

1.8.5.5 Estrategia de venta

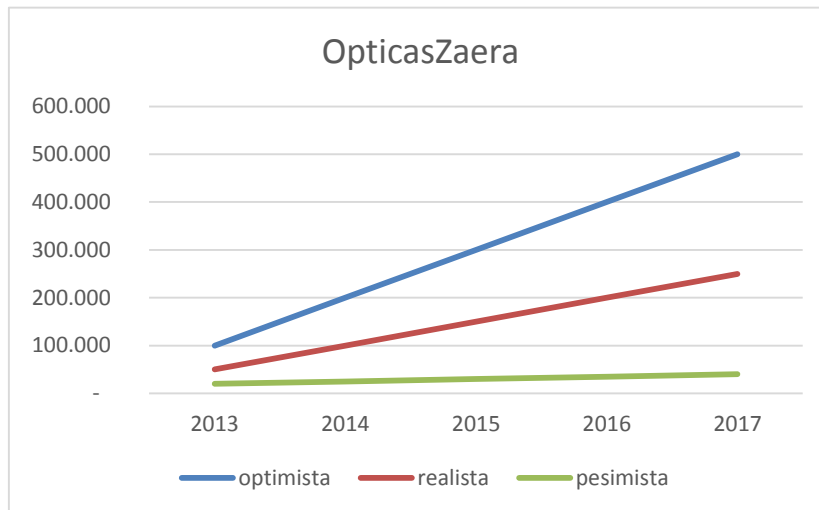
Para la estrategia de venta nos basaremos en la publicidad y en el marketing descrito anteriormente.

Estimaciones de venta:

Hemos realizado un análisis de como queríamos que nuestra empresa evolucionase a lo largo de los 5 años iniciales.

El siguiente análisis nos muestra cuales son la cantidad de producto que queremos vender, separándolo en tres escenarios: optimista, realista y pesimista.

Figura 4. Tres Escenarios futuros.



Fuerza de ventas:

Para poder tener un mayor aumento de ventas utilizamos técnicas que nos permiten gestionar las necesidades, comportamientos, indicadores de los clientes y de las ventas de los productos.

El sistema Google Analytics nos ayudara a tener un mayor volumen de ventas ya que nos hace una segmentación demográfica de los clientes. Además como para la compra de nuestros productos nuestros clientes se han tenido que identificar previamente, conocemos los datos de nuestros clientes y podemos realizar comparaciones que nos ayuden a conocer el sector sobre el que debemos de realizar un número mayor de ventas.

Esto nos permitirá poder obtener más ventas, por tanto un mayor beneficio para nuestro negocio.

1.8.5.6 Fidelización con los clientes

Para que nuestra empresa pueda obtener el mayor beneficio posible tendremos que realizar también una campaña de fidelización de los clientes para evitar que estos realicen compras y no se sientan del todo satisfechos y no vuelvan a realizar un pedido.

Al igual que mencionamos en la publicidad que íbamos a estar presentes en las redes sociales, para la fidelización del cliente también haremos uso de estas herramientas.

Las redes sociales mantienen un contacto muy directo con los clientes, por lo que podemos cuáles son sus inquietudes y que es lo que le gusta o desagrada de nuestra empresa.

También para una mayor comunicación con los clientes tendremos a su disposición un buzón de correo donde pueden dejar sus dudas, sugerencias y sus opiniones. Con esto podremos contestar a dudas relacionadas con las compras, productos, pedidos, etc, con el fin de que el cliente se sienta más cómodo sabiendo que hay una empresa seria y responsable que va a darle solución a sus problemas.

Nuestra tienda virtual está realizada mediante la herramienta de open source de Magento que cuenta con un sistema que te permite realizar comparaciones según las compras efectuadas lo que nos proporcionara un mayor conocimiento sobre cuáles son los productos más vendidos, sobre qué edad son usados, etc. Esto nos permitirá poder conocer más los gustos e intereses de nuestros clientes y así poder realizar la compra de productos que satisfagan a los clientes e intentar conseguir clientes potenciales.

1.8.6 Modelo de negocio

Propuesta de valor

El modelo de negocios será un B2C (Business to Consumer) y la página web tendrá un estilo de vendedores.

Modelo de ingresos

El modelo de ingresos será basado en ventas, como ya se explicó anteriormente, de venta de gafa, monturas, lentes de contacto y productos de limpieza ocular, que se le harán llegar a los clientes hasta su domicilio fiscal.

Oportunidad de mercado

En este momento en la región, serán todos los clientes, que se quieren aprovechar de nuestros precios, debido a la reducción de costes, que nos supone, este tipo de ventas, podemos ofrecer precios, más competentes.

Estudio de la competencia

En España existen de cientos de tiendas online gafas, pero pocas ofrecen todo tipo de productos y marcas, además de un servicio personalizado de graduación de cristales.

Ventaja competitiva

Al contar con infraestructura y tecnología de vanguardia, se lleva una gran ventaja sobre la competencia por ofrecer un servicio que ofrece la graduación ocular a través de la red, además supone una gran ventaja con nuestros grandes rivales, que en nuestro caso, son las

pymes, por amplitud de negocio, ya que pocas de ellas, disponen de estos tipos de servicios a través de internet.

Estrategia de marketing

Se invitará personalmente a todos nuestros clientes físicos de la tienda, para que conozcan todos los beneficios y controles que obtendrán en su negocio con el uso de este modelo de negocio por internet, también se les hará saber que pueden ellos mismos hacer del conocimiento a otros compañeros de negocios parecidos para se acerquen a la empresa y también conozcan las ventajas de este servicio de ventas.

Estructura organizativa

Dentro de la estructura organizativa se cuenta con dos contadores, dos auxiliares contables y un auxiliar técnico en sistemas para llevar el control de clientes en las ventas por internet.

Equipo de gestión

Se contará con un promotor que hará visitas periódicas en internet y visitas domiciliarias a los clientes para valorar que tan útil han sido las compras por internet y obtener propuestas para mejorar eficientemente con la atención y el servicio hacia los clientes.

Plan financiero

La inversión inicial, será 0, ya que se trata de un negocio familiar, que vamos a desarrollar nosotros mismos, y los productos, ya estaban en stock con anterioridad, con lo que los costes, serán los establecidos en la planificación del proyecto.

1.8.6.1 Opinión Modelo de Negocio

Se ha optado por este modelo de negocio, dado que es el más extendido por la red, y con el podemos llegar a un desarrollo mejor y más adecuado.

Por otro lado nos podemos decantar por una serie de ventajas según del lado que moremos (cliente/empresa).

Por el lado del cliente:

- La comparación de precios rápida y ágil.
- Opción de consultar información muy detallada.
- Sin necesidad de desplazamiento, compra a distancia.
- Lugar a compras más económicas.
- Precisa información del estado de sus compras.

Por el lado de nuestra empresa:

- La amortización a c/p.
- El coste de infraestructura es menor.
- Mejoras en la gestión del stock y no tener tanto nivel de inventario.
- Mayor aprovechamiento del recurso humano en área de compras.

- Expansión geográfica de mercado.

Bueno quizás estas ventajas que surgen son más generales, pero en resumidas cuentas, el modelo de negocio elegido es el adecuado, ya que es el modelo más usual de una tienda física. El concepto puede ser mejor asumido por los consumidores o incluso por nosotros mismos.

1.8.7 Plan de Proyecto. Diagrama de Costes

1.8.7.1 Objetivos

Nuestro objetivo es llegar cada año a más número de clientes, nuestro objetivo marcado es el siguiente:

Tabla 5. Objetivos a 5 años.

	A corto plazo (Año 1)	A medio plazo (Años 2-3)	A largo plazo (Años 4-5)
Facturación (€)	40.000	80.000	150.000
Pedidos / Año	20.000	40.000	65.000
Ámbito	Nacional	Nacional	Europeo

Por lo que la recuperación de la inversión realizada, sería a largo plazo, entre el 4º o 5º año.

1.8.7.1.1 Recursos con los que contamos

Contamos con todos los recursos, es una empresa que lleva funcionando más de 15 años, y tenemos suficiente stock de productos, como para hacer frente a esta expansión. Disponemos de todos los recursos, y para la publicación de la web, en un principio, se subirá a un servidor de hosting gratuito, por lo que tenemos los recursos necesarios, como para inicializar el proyecto.

1.8.7.1.2 Proveedores e intermediarios

-Proveedores de producto: Fabricantes / distribuidores

Nuestros proveedores de productos, continuarán siendo los mismos, y el mantenimiento de la página, la mantendremos nosotros, con nuestros propios recursos en un principio.

- Proveedores de servicios logísticos

Si el envío es urgente, se podrá hacer un envío desde 10 €, a través de correos, asumimos, que se enviarán de manera urgente, de manera poco frecuente. Para el resto de envíos contrataremos a un partner MRW, con el que negociaremos, un porcentaje de la cantidad enviada.

- Proveedores de alojamiento web

Para ello utilizaremos se trata de un servidor de hosting freiz.com gratuito que nos permite almacenar en la web hasta 1GB, y nos proporciona servicios MySQL y PHP. En un principio ya tenemos cuenta:

FTP/WEB: dhernanz.freeiz.com

U: a8940108

p: ****

```
$mysql_host = "mysql15.000webhost.com";
```

```
$mysql_database = "a8940108_a";
```

```
$mysql_user = "a8940108_a";
```

```
$mysql_password = "*****";
```

1.8.7.1.3 Diagrama de Costes/Presupuesto

El presupuesto vendrá marcado, por el precio de los recursos, y por el tiempo del proyecto, que en un principio, su estimación será la siguiente:

Costos de todo el proyecto:

Proyecto/Información del proyecto/Estadísticas

El coste Real previsto proyecto será de 142.240€, como se puede observar, ya hemos desembolsado 99.120€, por lo que la cantidad restante, es de:

Figura 5. Costos del proyecto.

Costos			
Programados:	142.240,00 €	Restantes:	43.120,00 €
Previstos:	0,00 €	Reales:	99.120,00 €
Variación:	142.240,00 €		

Por lo que el presupuesto restante a desembolsar es de 43.120€

Figura 6. Costos por recurso:

Nombre del recurso	Trabajo	Costo
--------------------	---------	-------

Jefe de Proyecto	464 horas	46.400,00 €
<i>Estudio de Factibilidad</i>	<i>8 horas</i>	<i>800,00 €</i>
<i>Estudio de Mercado</i>	<i>56 horas</i>	<i>5.600,00 €</i>
<i>Realizar Planificación</i>	<i>160 horas</i>	<i>16.000,00 €</i>
<i>Establecer requisitos</i>	<i>160 horas</i>	<i>16.000,00 €</i>
<i>Estrategia de Marketing</i>	<i>80 horas</i>	<i>8.000,00 €</i>
Analista Funcional	264 horas	21.120,00 €
<i>Estudio de Factibilidad</i>	<i>8 horas</i>	<i>640,00 €</i>
<i>Elaborar la Formulación</i>	<i>80 horas</i>	<i>6.400,00 €</i>
<i>Elaborar Diseño Arquitectónico</i>	<i>64 horas</i>	<i>5.120,00 €</i>
<i>Diseñar la interfaz</i>	<i>64 horas</i>	<i>5.120,00 €</i>
<i>Elaborar la Documentación Técnica</i>	<i>48 horas</i>	<i>3.840,00 €</i>
Analista Programador	464 horas	37.120,00 €
<i>Realizar Planificación</i>	<i>160 horas</i>	<i>12.800,00 €</i>
<i>Establecer requisitos</i>	<i>160 horas</i>	<i>12.800,00 €</i>
<i>Elaborar Diseño Arquitectónico</i>	<i>64 horas</i>	<i>5.120,00 €</i>
<i>Realizar Evaluación con el Cliente</i>	<i>80 horas</i>	<i>6.400,00 €</i>
Programador 1	224 horas	11.200,00 €
<i>Diseñar la navegación (Primera Entrega)</i>	<i>64 horas</i>	<i>3.200,00 €</i>
<i>Desarrollar Producción</i>	<i>80 horas</i>	<i>4.000,00 €</i>
<i>Generar Páginas de prueba</i>	<i>80 horas</i>	<i>4.000,00 €</i>
Programador 2	144 horas	7.200,00 €
<i>Diseñar la navegación (Primera Entrega)</i>	<i>64 horas</i>	<i>3.200,00 €</i>
<i>Desarrollar Producción</i>	<i>80 horas</i>	<i>4.000,00 €</i>
Tester	80 horas	4.000,00 €
<i>Generar Páginas de prueba</i>	<i>80 horas</i>	<i>4.000,00 €</i>
Soporte Hardware y Comunicaciones	0 horas	0,00 €
<i>Elaborar Diseño Arquitectónico</i>	<i>0 horas</i>	<i>0,00 €</i>
Experto en Base de Datos	80 horas	6.400,00 €
<i>Desarrollar Producción</i>	<i>80 horas</i>	<i>6.400,00 €</i>
Documentalista	48 horas	2.400,00 €
<i>Elaborar Documentación de Usuario</i>	<i>48 horas</i>	<i>2.400,00 €</i>
Experto en Prueba y Aseguramiento de Calidad	80 horas	6.400,00 €
<i>Generar Páginas de prueba</i>	<i>80 horas</i>	<i>6.400,00 €</i>

Figura 7. Costos por asignación:
Respecto a las tareas de Planificación y Diseño

	Total
Proyecto Ópticas Zaera	
Planificación	
Estudio de Factibilidad	1.440,00 €
Estudio de Mercado	5.600,00 €
Elaborar la Formulación	6.400,00 €
Realizar Planificación	28.800,00 €
Establecer requisitos	28.800,00 €
Diseño	
Elaborar Diseño Arquitectónico	10.240,00 €
Estrategia de Marketing	8.000,00 €
Diseñar la interfaz	5.120,00 €
Desarrollo	
Diseñar la navegación (Primera Entrega)	6.400,00 €
Desarrollar Producción	14.400,00 €
Pruebas	
Generar Páginas de prueba	14.400,00 €
Realizar Evaluación con el Cliente	6.400,00 €
Documentación	
Elaborar la Documentación Técnica	3.840,00 €
Elaborar Documentación de Usuario	2.400,00 €
Aceptación	
Comprar el Dominio (Segunda Entrega)	
Realizar Pagos Financieros	
Total	142.240,00 €

PLANIFICACIÓN:

La duración del proyecto, será de 96 días, y la planificación viene marcada por las siguientes tareas:

Tabla 6. Planificación del proyecto.

Nombre de tarea	Duración	Comienzo	Fin	Nombres de los recursos
Proyecto Ópticas Zaera	96 días	lun 25/02/13	jue 18/07/13	
Planificación	58 días	lun 25/02/13	lun 27/05/13	
Estudio de Factibilidad	1 día	lun 25/02/13	lun 25/02/13	Analista Funcional; Jefe de Proyecto
Estudio de Mercado	7 días	mar 26/02/13	mié 06/03/13	Jefe de Proyecto
Elaborar la Formulación	10 días	jue 07/03/13	mié 20/03/13	Analista Funcional
Realizar Planificación	20 días	jue 21/03/13	jue 25/04/13	Analista Programador; Jefe de Proyecto
Establecer requisitos	20 días	vie 26/04/13	lun 27/05/13	Analista Programador; Jefe de Proyecto
Diseño	26 días	mar 28/05/13	mar 02/07/13	

Elaborar Diseño Arquitectónico	8 días	mar 28/05/13	jue 06/06/13	Analista Funcional; Analista Programador; Soporte Hardware y Comunicaciones
Estrategia de Marketing	10 días	mar 28/05/13	lun 10/06/13	Jefe de Proyecto
Diseñar la interfaz	8 días	vie 21/06/13	mar 02/07/13	Analista Funcional
Desarrollo	18 días	mar 28/05/13	jue 20/06/13	
Diseñar la navegación (Primera Entrega)	8 días	mar 11/06/13	jue 20/06/13	Programador 1;Programador 2
Desarrollar Producción	10 días	mar 28/05/13	lun 10/06/13	Programador 1;Programador 2;Experto en Base de Datos
Pruebas	20 días	mar 11/06/13	lun 08/07/13	
Generar Páginas de prueba	10 días	mar 11/06/13	lun 24/06/13	Programador 1;Tester;Experto en Prueba y Aseguramiento de Calidad
Realizar Evaluación con el Cliente	10 días	mar 25/06/13	lun 08/07/13	Analista Programador
Documentación	6 días	mar 09/07/13	mar 16/07/13	
Elaborar la Documentación Técnica	6 días	mar 09/07/13	mar 16/07/13	Analista Funcional
Elaborar Documentación de Usuario	6 días	mar 09/07/13	mar 16/07/13	Documentalista
Aceptación	28 días	mar 11/06/13	jue 18/07/13	
Comprar el Dominio (Segunda Entrega)	2 días	mié 17/07/13	jue 18/07/13	
Realizar Pagos Financieros	2 días	mar 11/06/13	mié 12/06/13	

A continuación se muestra el diagrama de Gantt:

Figura 8. Diagrama de Gant.

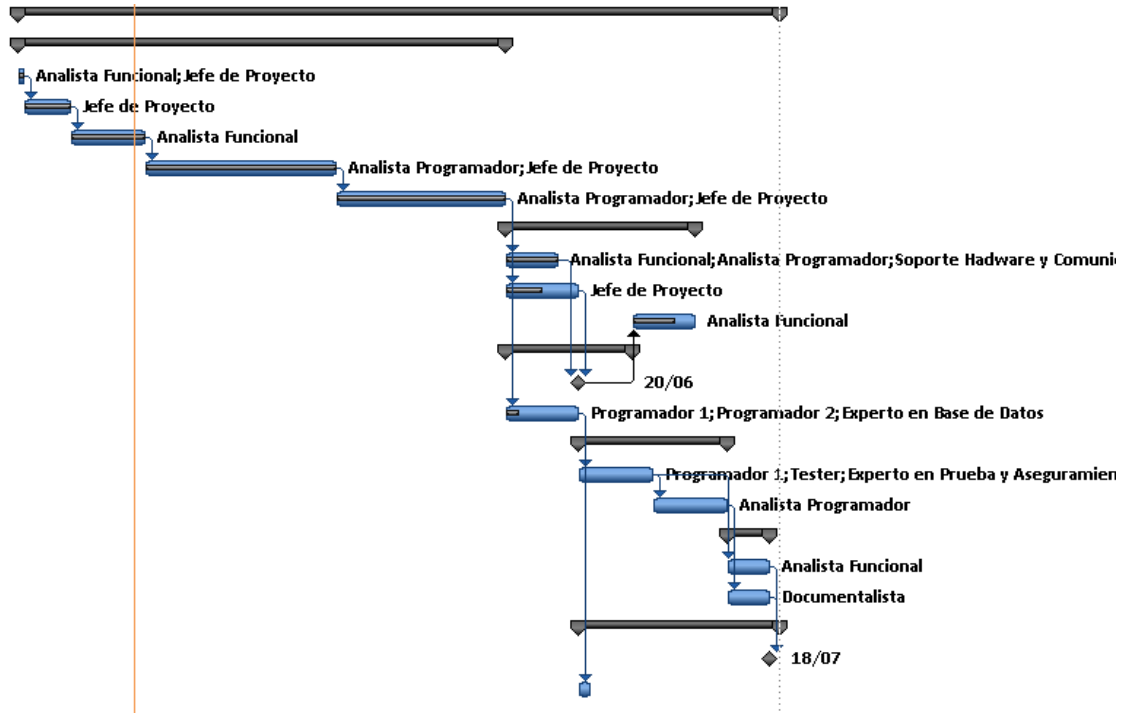
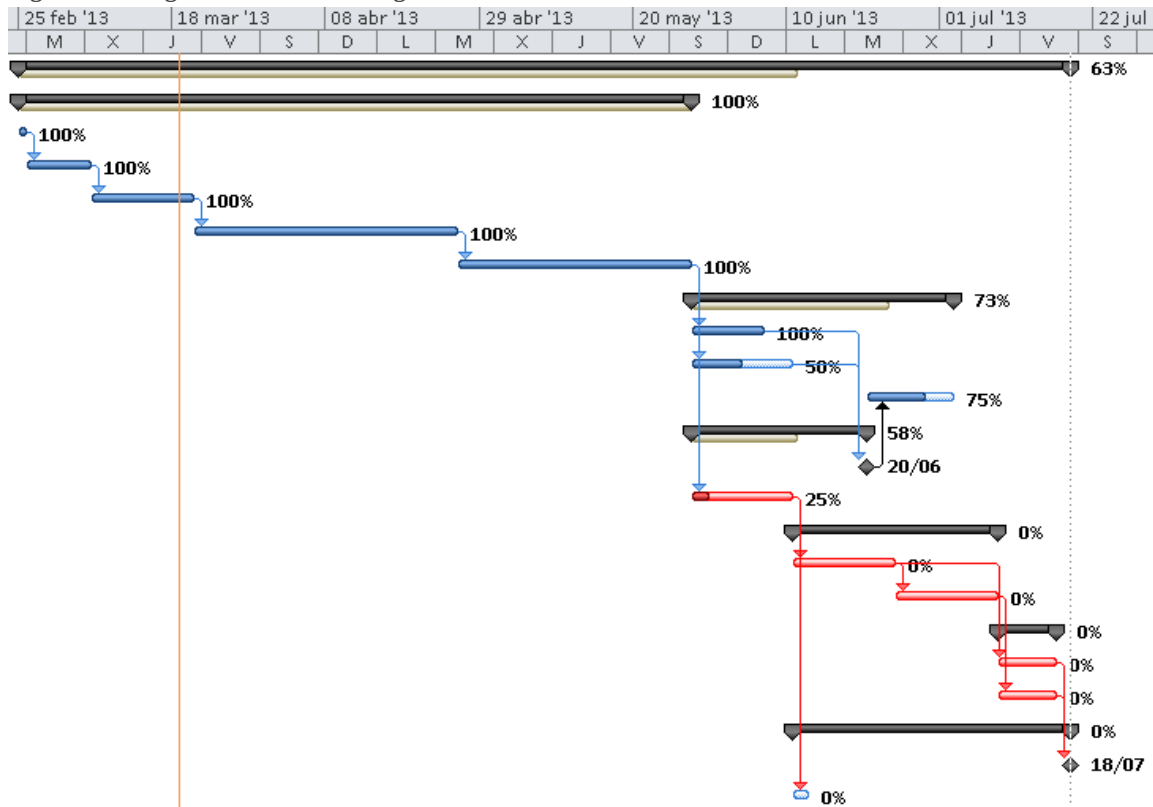


Figura 9. Diagrama de Gant desglosado:



Donde se puede observar las tareas completadas y el estado de cada una de las tareas, además de la planificación de dos hitos, que serían las 2 entregas del proyecto, una en fase pruebas, y la otra ya con la finalización del proyecto.

1.8.7.2 ORGANIZACIÓN Y RECURSOS HUMANOS

Tabla 6. Tasas por recurso humano.

Nombre del recurso	Tasa estándar
Jefe de Proyecto	100,00 €/hora
Analista Funcional	80,00 €/hora
Analista Programador	80,00 €/hora
Programador 1	50,00 €/hora
Programador 2	50,00 €/hora
Tester	50,00 €/hora
Soporte Hardware y Comunicaciones	100,00 €/hora
Experto en Base de Datos	80,00 €/hora
Documentalista	50,00 €/hora
Experto en Prueba y Aseguramiento de Calidad	80,00 €/hora

1.8.7.2.1 Organización funcional de la empresa

Jefe de Proyecto:

- Colaboración con el cliente en la definición y concreción de los objetivos del proyecto.
- Planificación del proyecto en todos sus aspectos, identificando las actividades a realizar, los recursos a poner en juego, los plazos y los costes previstos.
- Dirección y coordinación de todos los recursos empleados en el proyecto.
- Mantenimiento permanente de las relaciones externas del proyecto: clientes, proveedores, subcontratistas, otras direcciones, etc.
- Toma de decisiones necesarias para conocer en todo momento la situación en relación con los objetivos establecidos.
- Adopción de las medidas correctoras pertinentes para poner remedio a las desviaciones que se hubieran detectado.
- Responder ante clientes y superiores de la consecución de los objetivos del proyecto.
- Proponer, en su caso, modificaciones a los límites u objetivos básicos del proyecto cuando concurren circunstancias que así lo aconsejen.

Analista funcional:

Competencias Blandas

- Gestión del proyecto en general
- Comunicación interpersonal
- Comprensión de procesos empresariales
- Conocimiento de la Organización
- Capacidad de Negociación
- Adaptación al Cambio
- Investigación y Proactividad

Competencias Técnicas

- Conocimiento de la herramienta
- Configuración del Proceso
- Identificación de Roles y Perfiles
- Migración de Datos
- Especificaciones del Desarrollo

Analista Programador:

Competencias Blandas

- Pensamiento lógico
- Interés por el orden
- Constancia
- Capacidad de atención y concentración
- Innovación

Competencias Técnicas

- Paradigma de Objetos
- Lenguaje de diagramación de sistemas UML
- Lenguaje de consulta de bases de datos SQL
- Conocimiento real de al menos un lenguaje de programación
- Técnicas de calidad de software.

Diseñadores:

- Especificaciones del software.
- Análisis de los requerimientos del software.
- Diseño y mejora de prototipos y de demos para validar requerimientos.
- Testeo de las aplicaciones y supervisión del proceso de arranque de la aplicación.
- Mantenimiento.

Programadores: Se encargarán de todas las funciones de tratamiento y desarrollo de código.

Tester: se encargará de las pruebas.

Soporte HW y comunicaciones: persona encargada de todo el hardware de la empresa junto como el sistema de comunicaciones, y de todo el mantenimiento de ello.

Expertos en Bases de Datos:

- Administrar, mantener y diseñar bases de datos con MYSQL.
- Aprender a manejar bases de datos con el lenguaje de consultas SQL.

Documentalista:

- Ser capaz de autenticar y validar las fuentes y los recursos.
- Saber manipular los documentos electrónicos para ponerlos a disposición de los usuarios, cualquiera que sea su plataforma de trabajo.
- Poder realizar nuevas operaciones de almacenamiento, archivo y capitalización activa de lo que es pertinente y útil con la idea de poder re explotarlo.

- Concebir, crear y realizar nuevos productos y servicios que pongan de manifiesto las potencialidades de Internet, que satisfagan a los usuarios y que valoren las competencias de los documentalistas.
- Ayudar a los usuarios en las funcionalidades y recursos de Internet.
- Organizar nuevas redes de acceso a la información, la documentación y el conocimiento.
- Definir e implementar metodologías para la determinación del conocimiento necesario en el cumplimiento de los objetivos, desarrollando formas de difusión de los conocimientos y su reutilización.
- Determinar la información y el conocimiento necesarios para el desarrollo tanto de los procesos como de los procedimientos de actividad.
- Detectar e integrar la información externa con la producida en la propia organización.

1.8.7.2.2 Plan de Recursos Humanos y perfiles requeridos

Además de los perfiles mencionados anteriormente ante la casuística de la toma de decisiones se realizarán en el pleno consenso. Cuando el pleno consenso no pueda ser alcanzado, para evitar los estancamientos en dicho proceso de toma de decisiones primará la palabra final de:

- Mario Gutiérrez: Cuando los temas sobre los que se debate estén relacionados con política comercial, de atención al cliente, uso de redes sociales, promociones o campañas especiales y por ultimo asuntos técnicos de software del espacio Web.
- Primer socio: Cuando Los temas sobre los que se debate estén relacionados con la estética del espacio web, organización de sus contenidos y todo lo relacionado con aspectos financieros de la empresa.
- Segundo socio: Cuando los temas sobre los que se debate estén relacionados con marketing,
- Para los demás temas en los que se debatiera y que no se encuentren reflejados en los tres puntos anteriores, prevalecerá la decisión que considere oportuna Mario Gutiérrez.

1. Condiciones remunerativas:

Del obtenido se creará una reserva del 5% a 30% que será destinada al patrimonio de la sociedad. Se elegirá por consenso según los resultados y las expectativas del año siguiente.

El dinero previsto para los sueldos de los promotores será el mínimo interprofesional el primer año esperando aumentarlo año tras año pero sin llegar a ser excesivo y pudiendo utilizar el propio sueldo para la propia empresa si se necesitara.

2. Plan de formación:

Cada año se recurrirá a un 3% del patrimonio de la sociedad. Para subvencionar un curso de formación relacionado con la actividad de la empresa a un socio distinto. Si el curso elegido sobrepasara en prestación económica el socio correrá con el excedente de gasto.

1.8.7.2.3 Previsión de Recursos Humanos

Nuestra previsión para los próximos años es la siguiente:

- Si la facturación anual llegara 200.000 euros procederíamos a la contratación de 1 persona esperamos llegar a este objetivo en 2014.
- Si la facturación anual llegara a 350.000 euros procederíamos a pasar de 1 trabajadores a 4.
- Si la facturación anual llegara a 500.000 euros procederíamos a pasar de 4 trabajadores a 6.
- Si la facturación anual llegara a 750.000 euros procederíamos a tener un equipo de 15 trabajadores. Distribuyendo el equipo según precise en los departamentos correspondientes.

Nuestro código moral como empresa:

En general nuestra empresa no será donde – además de los salarios – se ofrezcan grandes compensaciones complementarias, somos de la opinión de que la recompensa por el trabajo bien hecho es una excelente remuneración y que todo lo demás logra incrementar los gastos de la empresa pero no los rendimientos de los trabajadores.

Por tanto nuestra concepción general es la siguiente:

- Todos los colaboradores de la empresa deben percibir un salario fijo y digno (pero nunca espectacular) en función de sus responsabilidades en la compañía y el mercado.
- El equipo directivo deben tener la oportunidad de obtener excelentes remuneraciones (bonos) complementarias vinculadas a sus resultados y-o a los propios resultados de la empresa.
- El personal de marketing, y sobre todo de ventas deben estar excelentemente remunerados pero el 80% de dicha remuneración debe obtenerse por el logro de sus objetivos personales (80%) y generales (20%).
- La empresa no prestará compensaciones de ningún otro tipo que las obligatorias por ley, salvo en los casos justificados operativamente; No habrá cuentas de gastos, ni similares.

1.8.7.3 ASPECTOS LEGALES Y SOCIETARIOS

1.8.7.3.1 La sociedad y Sede social/operativa

La empresa constará de:

Sociedad de Responsabilidad Limitada, repartido a partes iguales:

- Primer socio
- Segundo socio
- Mario Gutiérrez Calvo de Mora

La sede se encontrará en:

Apartado de Correos nº 129
15895 Milladoiro, Ames
A Coruña

1.8.7.3.2 Licencias y derechos

Una vez puesta en marcha la tienda online, se registrará en un dominio .es, dedicándose en un principio únicamente al territorio español.

La marca ha sido registrada en la “Oficina de Patentes y Marcas” con el nombre de “Ópticas Zaera” y con el siguiente logo:

Figura 9. Logo.



1.8.7.3.3 Obligaciones Legales

Cumplimiento y obligación de:

- La LOPD (Ley orgánica de protección de datos).
- Ley de Servicios de la Sociedad de Información y Comercio Electrónico (LSSICE)
- Licencias legales de software, exceptuando Magento (software libre).

Por otro lado:

Los ópticos en su ejercicio profesional estarán obligados a compartir la historia clínica de sus pacientes con otros profesionales, para unificar criterios de actuación basada en la evidencia científica regulados en protocolos de práctica clínica.

Tendrán que conseguir un uso racional de los recursos diagnósticos y terapéuticos, tomando consciencia de sus decisiones, y evitando una inadecuada utilización de los mismos.

Todos los ópticos tienen que respetar las decisiones del paciente después de informarle de todas sus opciones de tratamiento.

Estarán obligados a redactar informes de valoración y de resultados de todos los tratamientos que realicen a sus pacientes.

1.8.7.3.4 Permisos y licitaciones

Permisos sanitarios referentes al Real Decreto 14/2003 de 13 de febrero, referentes a la puesta en marcha de una óptica.

Las licencias pertinentes para el almacén y la licencia del funcionamiento de la tienda online.

1.8.7.4 ESTABLECIMIENTO, LANZAMIENTO E INVERSIONES.

1.8.7.4.1 Estado del proyecto y Plan de Establecimiento

- Desarrollo: Partimos de una base ya desarrollada por “Ópticas Zaera”, nos encontramos en estos momentos terminando de desarrollar el sitio web y las funcionalidades de la tienda online.
- Formación del personal inicial: La formación inicial se está realizando.
- Infraestructuras: por el momento disponemos del local donde se encuentra la tienda física, una oficina y un almacén.

1.8.7.4.2 Plan de Lanzamiento

El objetivo de nuestro plan especial de lanzamiento es obtener notoriedad desde el primer minuto. No queremos ser una empresa más, queremos hacernos notar en todos los ámbitos en el menor tiempo posible y para ello desarrollaremos una suma de acciones y mini-campañas que pretendan generar un efecto positivo para nuestros intereses. Nuestros objetivos son:

- Hacernos notar: que las empresas de nuestro entorno nos conozcan con la mayor brevedad.
- Posicionarnos desde el primer momento como una compañía muy relevante, que aporta calidad y valor añadido.

1.8.7.5 PLAN FINANCIERO

Necesidades financieras: Este proyecto requiere un volumen de financiación global que hemos establecido en los siguientes dos grupos.

1. Financiación de establecimiento: 88.500 €

2. Financiación de lanzamiento: 69.000 €

Necesidad total de fondos 157.500 €

- Plan de financiación: Por razones de prudencia y a pesar de nuestra absoluta confianza en el negocio y el cumplimiento de los objetivos, hemos previsto cubrir todas las necesidades mediante capital aportado por los socios.

- Capital aportado: Hemos aportado un capital por los socios 98.000 € aportaran los socios de la empresa. Dicho capital cubrirá el 62% de las necesidades totales.

Socios: El resto de financiación necesaria se prevé mediante una asociación de 59.000€. Dadas las perspectivas y los márgenes previstos, dicha financiación se considera suficiente y recuperable por nuestros socios.

2 Capítulo 2: Desarrollo/Implementación

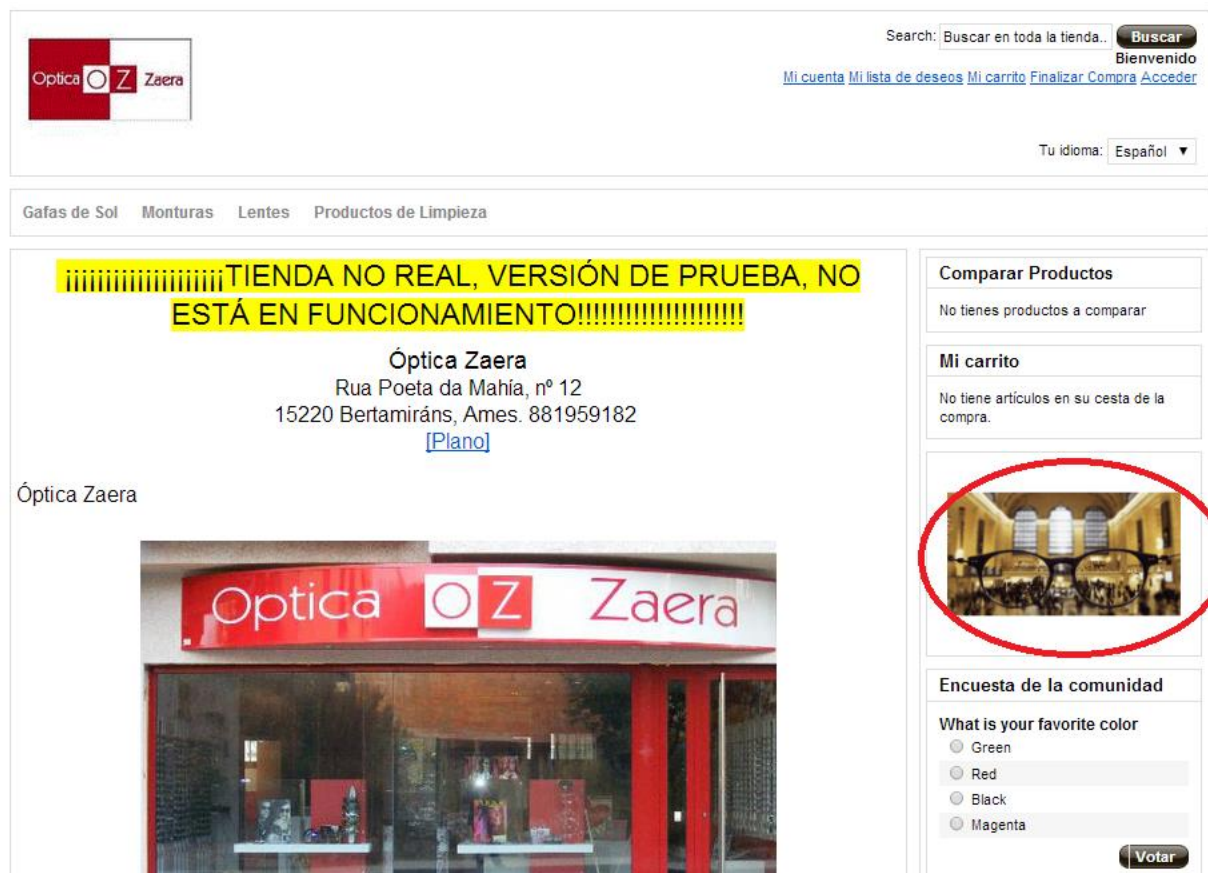
2.1 File system Magento

Para cambiar la imagen que sale resaltada, hay que ir a la siguiente ruta:

```
/public_html/app/design/frontend/base/default/layout
```

Dentro de esta ruta tenemos los siguientes archivos xml, donde se definen los diferentes bloques de la web. El de la siguiente imagen se encuentra dentro del archivo "Catalog.xml".

Figura 10. Vista Original tienda.



3 Desarrollo/Implementación basado en WebKit

(Probar varias herramientas, y solo aportar las que funcionan bien) Si fuera poco y fácil: analizar herramientas que existen (de pago y gratis) compararlas y hacer un estudio.

Un aspecto que cada día cobra más relevancia, es el de diversificar el negocio incorporando la funcionalidad de vender mediante plataformas móviles. Esto se conoce como m-Commerce.

El origen de este tipo de negocio nace en los primeros servicios móviles de mensajes de pago y de iniciativas de entidades de utilizar sistemas de pago a través del móvil como: CaixaMóvil, Paybox, MobiPay, etc.

Durante los últimos años, la llegada y crecimiento de la utilización de dispositivos móviles inteligentes o llamados Smartphones ha sido inminente, en todo tipo de personas. Lo que ha provocado un aumento significativo de las actividades comerciales a través de los mismos.

Según un estudio realizado por Google denominado “Our Mobile Planet: Global Smartphone User” (Nuestro planeta móvil: Usuarios de smartphones globales), el cual estudia la tendencia de países como: Estados Unidos, Reino Unido, Francia, Alemania, Japón y España, el cual es el 2º país con mayor inferencia de estos dispositivos con el 44%, teniendo únicamente por detrás a Reino Unido con el 45%.

Los usuarios que utilizan sus dispositivos al realizar una búsqueda de información y que lo hacen mediante un anuncio de publicidad convencional o de una tienda, tiene un porcentaje del 68%, pero por otro lado, no resaltamos al realizar compras a través de nuestros dispositivos móviles. Por ejemplo, en Japón más del 40% de los usuarios emplean los dispositivos móviles para comprar.

3.1 WebKit: Indiferencia entre iOS y Android

Es una plataforma Open Source para aplicaciones, la cual funciona como base diferentes navegadores. Basada originalmente en el motor de renderizado KHTML del navegador web del proyecto KDE, Konqueror.

Este proyecto consta de tres módulos diferentes:

- JavaScriptCore: es un motor para JavaScript.
- WebCore: el cual maneja el código HTML y las plantillas CSS.
- WebKit: es una API para relacionarse con el navegador.

La principal ventaja de este motor, a parte de la compatibilidad entre diferentes navegadores, y por ello entre diferentes dispositivos; se basa en la gran ayuda a la hora de construir Aplicaciones Web, basar las app que se crean para los diferentes smartphones.

3.1.1 WebKit y HTML5

La mayoría de los usuarios, lo que buscan es la integración y que el contenido web que visitamos sea accesible, que no es lo mismo que tener accesibilidad a nuestros datos independiente del lugar donde nos encontremos (movilidad).

La especificación HTML5, abarca un conjunto de proyectos técnicos para cubrir una gran variedad de tecnologías que toman como base el navegador, incluyendo en ello el cliente SQL, el cual se encarga del almacenamiento de todos los datos, transformación, las transacciones, traducción y muchas más cosas.

3.2 Consideraciones sobre la aplicación web Móvil

Las principales características que abarca el usar una aplicación web móvil, es que resuelve muchos de los problemas que tradicionalmente afectan a los desarrolladores de aplicaciones y los administradores de sistemas. Algunas de las cosas que resuelve o facilita WebKit son:

- La facilidad de implantación y puesta en marcha, es tan fácil como subir la aplicación al servidor y redireccionar a la URL correcta.
- Fácil escalabilidad.
- Facilidad para centralizar el almacenamiento de datos, lo que simplificaría los backups.
- La unión de los tres elementos más importantes:
 - CSS, hojas de estilo en cascada.
 - JavaScript e imágenes gráficas.
 - Y por último, y el más importante, la implementación con HTML.

Hace que la experiencia de interfaz de usuario sea impresionante, superando las capacidades que pueda tener un SDK nativo.

- Interfaces de usuario que sean intuitivas y fáciles de usar.

El añadido de incluir servicios externos en una app web, como pueden ser sistemas de pago, el simple pago con una tarjeta de crédito o PayPal, cualquier transacción a distancia. Los proveedores de software se benefician porque el desarrollo/actualización de estos servicios se pueden simplificar ampliamente, reduciendo costes de soporte y por consiguiente los costes transferidos al cliente. El modelo de software como servicio (SaaS, los datos y el soporte lógico lo maneja una empresa TIC), permite a los clientes minimizar la inversión inicial, obteniendo un ROI (Retorno de la Inversión) mucho más próximo.

Existen dos tipos básicos de presentación de aplicación web adaptadas a un dispositivo móvil:

3.2.1 Presentación Genérica del sitio

El motor de renderizado Webkit, junto con la interfaz de usuario intuitiva que se encuentra en las diferentes plataformas, permite visualizar cualquier sitio Web que esté basado en HTML. Cuando la página se cargue, el contenido se ampliará ajustándose al dispositivo, aunque quede demasiado pequeña, pero de esta forma se verá entera de un primer vistazo, ampliando el usuario según la parte que quiera ver de ella. De modo contrario, la página aparecería sin ajustar y cargando inicialmente solo una pequeña parte de ella, teniendo con el usuario una primera toma de contacto desagradable. Todo esto es un paso hacia delante para mejorar la experiencia y primera toma de contacto con el usuario.

3.2.2 Mobile-friendly

El siguiente paso es avanzar en la modificación de algunas áreas para seguir mejorando. No es suficiente que la página se muestre correctamente en WebKit, ya que existen algunas diferencias a la hora de mostrar la Web desde un pc a un dispositivo móvil táctil.

Las principales áreas de divergencia incluyen el tamaño físico de las áreas “hacer clic”, la falta de “estilos” y una secuencia de eventos totalmente diferente. A continuación, un resumen de las principales cosas a tener en cuenta para diseñar un sitio Web al alcance de los usuarios móviles:

- Sacar el máximo partido a la versión móvil, ya que la pantalla será mucho más legible.
- Los dedos son más grandes que los punteros del ratón. Hay que tener cuidado de no colocar enlaces demasiado juntos, ya que el usuario tendrá cierta dificultad para pulsar sobre ellos.
- Los Estilos Hover no funcionan porque no lo hace “flotar” con el dedo de la misma manera que se haría con el ratón.
- Tener en cuenta que los eventos como el ratón abajo, el ratón-mover, etc son diferentes en los dispositivos táctiles. Algunos de estos eventos se disparará, pero no se puede confiar en que surja el mismo efecto que en el navegador de un PC.

Para nuestros propósitos, vamos a centrarnos más en lo que WebKit puede hacer, en lugar de lo que no puede.

El primer desafío que se presenta es que el sitio web se adapte a la pantalla del dispositivo móvil, tanto si se visualiza con el dispositivo en posición vertical, como si este se presenta de forma horizontal. Webkit hace que las páginas web de escritorio se muestren y adapten muy bien, aunque el texto podría ser demasiado pequeño para que pueda ser leído de forma eficaz sin tener que pellizcar para hacer zoom. Entonces, lo que podemos hacer al respecto es emplear una meta etiqueta, la manera más eficaz podría ser acomodar la página web al dispositivo móvil a través del uso la metaetiqueta especial “viewport”.

Una metaetiqueta es una etiqueta HTML colocada dentro del elemento Head de un documento HTML. Un ejemplo simple del uso de viewport sería:

```
<meta name="viewport" content="width=device-width" />
```

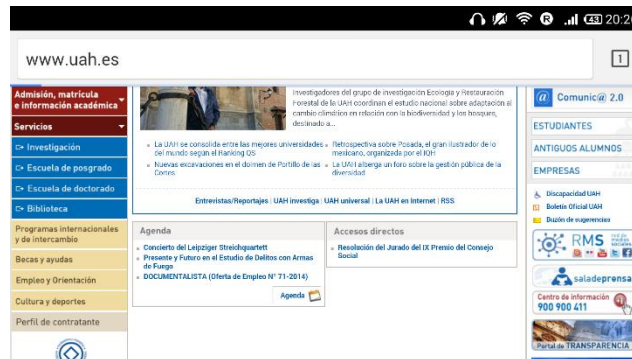
Cuando se añade este metatag a una página HTML, vemos que la página se adapta a la escala más adecuada para el escritorio móvil. El navegador que no lo soporta, simplemente ignora esta etiqueta. Un ejemplo sería la misma página de la universidad, en el primer caso se ajusta de tal forma:

Figura 11. Ejemplo web original adaptada al dispositivo.



Y si giramos el dispositivo de forma horizontal, se puede observar que vuelve a ajustarse con los nuevos valores:

Figura 12. Ejemplo web original adaptada al dispositivo en horizontal.



Hay veces, que es interesante preestablecer una escala específica, el contenido del atributo metatag viewport se establece en un valor explícito, un ejemplo de ello podría ser:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0 user-scalable=yes" />
```

Al cambiar el valor de la escala inicial, la pantalla puede hacer zoom o alejarse si se desea. Lo ideal es que el valor de escala inicial este entre 1.0 y 1.3. Con este metatag también se pueden considerar los valores máximo y mínimo de la escala, para limitar el control que el usuario pueda tener sobre la representación de la página. Para la App web de la tienda, se va a emplear incluyendo la siguiente línea:

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
```

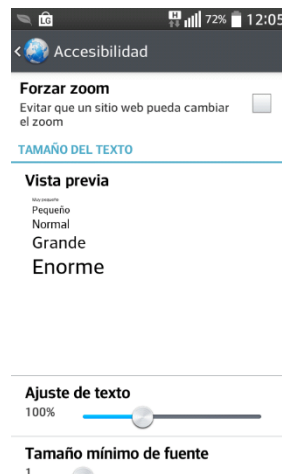
Dentro del siguiente archivo:

```
"\magento\app\design\frontend\default\iphone\template\page\html\ head.phtml"
```

La potencial variabilidad en el tamaño de pantalla, factor de forma y la resolución, es algo a tener en cuenta al desarrollar las aplicaciones y orientar los dispositivos móviles como Android.

El software de Android ha ejercido un mayor control y tiene una principal ventaja, que es la presentación de las páginas a través de la configuración del navegador en el dispositivo. La plataforma Android, además de estable es muy fluida. La configuración particular de un dispositivo, difiere seguramente de su entorno de desarrollo, en función del nivel de SDK y el fabricante. La pantalla de configuración permite al usuario configurar un dispositivo a un nivel de zoom predefinido (lejos, cerca y medio) o para solicitar que el dispositivo ajuste de forma automática la página. El siguiente ejemplo es sobre las opciones de una versión de Android 4.4.2:

Figura 13. Configuración navegador Android



En el mundo de los móviles, la mayor ventaja es el constante cambio, por lo que promete ser un entorno variable. Uno de los ejemplos es la configuración del navegador “Sprint Hero”, el cual está dotado de un conjunto totalmente diferente de opciones en relación a la presentación de las páginas. Afortunadamente, estos ajustes serán sustituidos por el metatag viewport que se ha explicado anteriormente, si está presente en la página Web.

Hasta ahora, todo lo que se ha visto sobre WebKit es hacer una página Web normal, bastante bien, aunque de pequeño tamaño y sin profundizar excesivamente sobre ella. A continuación, se ejerce el control sobre la experiencia de usuario, a través del uso del metatag viewport, se toman las riendas de como se ve la página en el dispositivo. Esto hace que la página sea más fácil de ver y navegar. Pero ¿y si se quiere ir más allá y hacer que el sitio pueda verse e integrarse como una aplicación móvil?

3.2.3 Hecho para móviles

Ahora se va a echar un vistazo a una estrategia de diseño cuando se dirigen a un público móvil. Como caso de estudio rápido, se toma como ejemplo, la página de acceso a Facebook.

En primer lugar, se muestra el diseño para un explorador de escritorio de un PC:

Figura 14. Diseño explorador escritorio.



La pantalla de inicio de escritorio tiene contenido informativo a la izquierda y un área de inicio de sesión a la derecha.

En segundo lugar, se contrasta el diseño para un explorador funcionando en plataforma iOS con la imagen anterior:

Figura 15. Diseño explorador móvil.



La pantalla que se muestra en la segunda imagen, sin duda va dirigida a un usuario móvil. Se solicita al usuario directamente lo necesario para seguir adelante con la aplicación, sin obligarle a tener que hacer zoom o desplazarse más de lo necesario.

A continuación, vamos a ver la funcionalidad de la aplicación Facebook para móviles al navegar. Debido al espacio, limitado y reducido tamaño para la aplicación, la ventana de lectura tiene muy poca oportunidad de mostrar otros elementos como botones, combos, etc. Cualquier espacio que se ocupe con otros elementos y esté dedicado a la navegación, reduce el contenido de lectura. Además, se debe evitar tener varios elementos div de desplazamiento. Mobile Facebook resuelve este problema proporcionando un sencillo menú en la parte superior con las funciones más comunes.

Figura 16. Diseño explorador móvil 2.



Y en la parte izquierda, si se pulsa se despliega otro menú mucho más extenso desde la parte izquierda, como se puede observar en la siguiente figura:

Figura 17. Diseño explorador móvil 3.



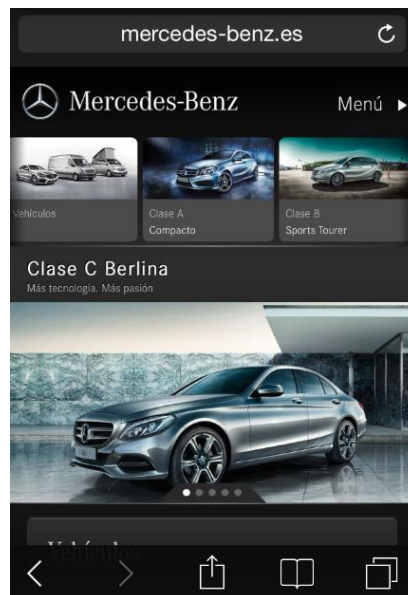
Como se puede observar es una aplicación web orientada por completo a los usuarios móviles.

Hay que tener en cuenta, a ser posible, que aunque el usuario acceda desde un dispositivo móvil, no debemos privarle de poder visualizar la Web en su formato de escritorio, con este fin se incluyen los siguientes enlaces para que el usuario acceda a la versión que desee solo con pulsar sobre ella.

Si el usuario prefiere la mayor funcionalidad de la versión de escritorio, que elija la opción que prefiera. Ahora, digamos que la página se quiere construir utilizando tecnologías Web, pero en realidad se vea como una aplicación nativa.

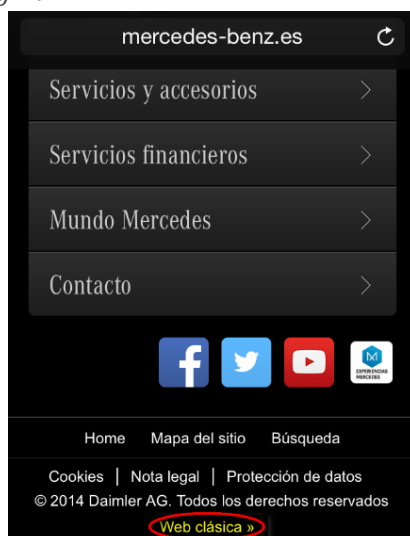
En el caso de Facebook no contempla esta opción, pero hay otras muchas páginas que si, como por ejemplo la página de mercedes-benz:

Figura 18. Versión móvil.



Con el enlace abajo del todo, para que se visite la versión original si el usuario lo desea:

Figura 19. Enlace versión original.



3.2.3.1 El contenido específico de la plataforma

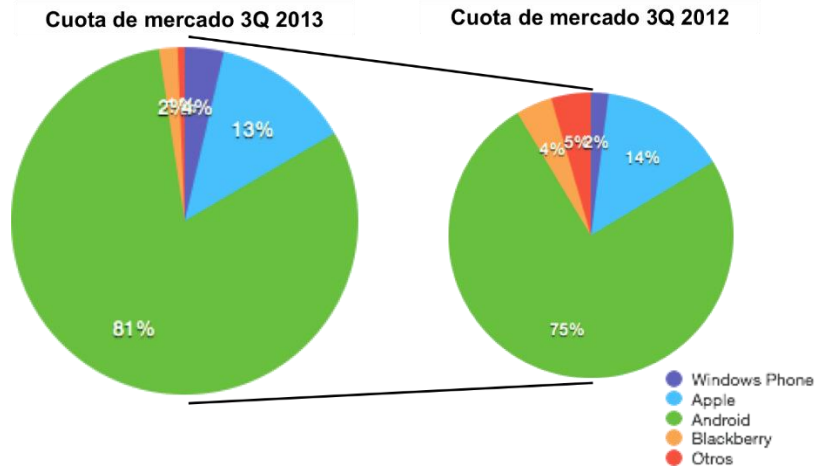
El siguiente paso es crear contenido específico de la plataforma orientado al formato de una página mirando más de cerca al comportamiento nativo para la plataforma destino, en lugar de un sitio Web genérico. Lo que se quiere decir como nativo es lo siguiente:

Antes de profundizar en los detalles de cómo desarrollar una página Web visual y que se esta se sienta como una aplicación nativa para una plataforma en particular, vamos hacer una breve comparación en las diferencias visuales entre iOS y Android como plataformas, dejando de lado por el momento su fuerte relación basada en el navegador.

- iOS tiene un aspecto distinto. Si se muestra a alguien una captura de pantalla de un iPhone/iPad y a menos que no se hayan sumergido lo más mínimo en el mundo de los dispositivos móviles, casi al 100% reconocerá la imagen proveniente de un dispositivo móvil de Apple. Sin embargo, si se muestra a la misma persona una captura de pantalla de un dispositivo Android, el resultado probablemente será diferente. ¿Por qué ocurre esto? La razón principal es que ha estado en el mercado por más tiempo, y forma parte de un culto de seguidores. Solo hay que volver la vista atrás, y ver a esa gente haciendo cola durante largas horas para pagar mucho dinero por un nuevo producto de Apple.
- Por otro lado, aunque Android ya lleva años en el mercado y se introdujo en el mercado de forma posterior, abarca prácticamente toda la comunidad OpenSource.

Ahora podemos decir que el número de dispositivos Android es sustancialmente mayor a los de iOS, podemos observar según la siguiente imagen que según las estadísticas:

Figura 20. Estadísticas usuarios móviles.



Esto se debe a que los dispositivos montados sobre la plataforma Android están siendo producidos por muchos fabricantes, y se emplean en todas las redes de operadores del mundo, por el contrario, aunque ya no tanto, iPhone ha intentado mantener muchas veces la exclusividad de cara a las redes de operadores móviles. Con el mayor número de usuarios que se encuentran dentro del mercado Android. Además, la apariencia visual no es universal dentro de esta plataforma, difiere muchas veces del dispositivo/fabricante que se posea, disponible algunas versiones solo para algunos.

A continuación, se expone donde se podría dar un aspecto nativo:

- Sitios totalmente paralelos.
- Los servidores proxy que filtran el contenido hasta el dispositivo. Por ejemplo, RIM en su día, hizo un uso extenso y exitoso de este enfoque para la renderización de correo electrónico en el dispositivo.
- Contenido generado dinámicamente a base de userAgent.

Estas tres características están más bien enfocadas para grandes equipos bien financiados. Para el resto, no hay tiempo, capacidad ni recursos suficientes para hacer frente a esto. Como se ha ido viendo, la web móvil de hace un tiempo fue insuficiente.

Por suerte, para la era de WebKit y CSS, estas diferencias se pueden abarcar fácilmente mediante las hojas de estilo y otras características. Una “media query” es una técnica para obtener información sobre el cliente. Normalmente, los navegadores envían una cadena “userAgent”, que identifica el navegador y el servidor haciendo el trabajo de averiguar que contenidos va a enviar al dispositivo (según se ha explicado más arriba). Con una consulta de “media query”, el navegador toma la decisión en base a sus capacidades. A continuación, se muestra un ejemplo para la obtención de una hoja de estilo dirigida a un dispositivo inteligente:

```
<link rel = tipo "hoja de estilo" = "text / css" href = "smartphone.css" = media "sólo pantalla y (max-device-width: 480px)" />
```

Y lo siguiente está dirigido hacia una consulta de medios de un sistema de sobremesa:

```
<link rel = tipo "hoja de estilo" = "text / css" href = "smartphone.css" = media "sólo pantalla y (min-device-width: 481px)" / >
```

3.2.4 La construcción de la aplicación

Hoy en día las páginas web deben ser creadas de manera declarativa, que proporciona organización y contenido único. Todo el posicionamiento y el formato, se lleva a cabo a través de las hojas de estilo en cascada y, a menudo, con la ayuda de JavaScript. De hecho, las bibliotecas de JavaScript se han vuelto tan populares, que más que una Excepción, se están convirtiendo en una norma. En aplicación de este estudio se aprovecha la infraestructura popular jQuery JavaScript. La aplicación de ejemplo se representará en iOS, Android, y también en un PC. El contenido HTML es el mismo, las diferencias se encuentran en las hojas de estilo implementadas para cada uno. Primeramente no se va a centrar en la apariencia de la aplicación, los colores de fondo son muy exagerados para hacer hincapié sobre la organización de estilos de la solicitud. El HTML se muestra a continuación:

Listado 1. HTML para la aplicación

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta name="viewport" content="width=device-width" />
<link rel="stylesheet" href="netmon.css" type="text/css" />
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="netmon.js"></script>

<script type="text/javascript">
  if (navigator.userAgent.indexOf('iPhone') != -1) {
    document.write('<link rel="stylesheet" href="iphone.css"
type="text/css" />');
  } else if (navigator.userAgent.indexOf('Android') != -1) {
    document.write('<link rel="stylesheet" href="android.css"
type="text/css" />');
  } else {
    document.write('<link rel="stylesheet" href="desktop.css"
type="text/css" />');
  }
}

function setupTestData() {
  try {
    netmon.initialize();
    if (netmon.resources.length > 0) {
      jQuery.each(netmon.resources,function (index, value) {
        $("#mainContent").append(netmon.render(index,value));
      });
      $(".serverentry").click(function() {$(this).find(".serveritems").toggle();});
      $(".serveritems").hide();
    }
  } catch (e) {
    alert(e);
  }
}
</script>

<title>My Network Resources</title>
</head>
<body onload="setupTestData();">
<div id="mainContainer">
```



```
<div id="header">
  <h1>My Servers</h1>
</div>
<div id="mainContent">
</div>
  <a href="q.php">My User Agent</a>
</div>
</body>
</html>
```

De un vistazo rápido al código HTML, podemos observar varias cosas como las siguientes:

- Gracias al uso del *Viewport metatag* ajusta el contenido renderizando a escala.
- Hay dos archivos *load JavaScript*: uno de la librería de jQuery y otro de funciones de ayuda para la aplicación.
- Se carga una primera hoja de estilo "*netmon.css*".
- El *userAgent* pregunta para determinar cuál es la hoja de estilo adicional para cargar: *iphone.css*, *Android.css* y una para escritorio *Desktop.css*.
- Cuando la página es cargada, se carga la ayuda de jQuery y una función de ayuda en el archivo *netmon.js*.
- El balance de la página contiene una pareja de *div tags*.
- Por último, hay un link con una página que muestra el string del *userAgent*. Esto es un lugar solo para las propuestas convenientes.

Antes de bucear en las hojas de estilo y el archivo *netmon.js* donde todo el trabajo está hecho, se va a dar un vistazo a la solicitud en su forma actual. Es de recordar que se están usando tres hojas de estilo dirigidas a tres plataformas diferentes soportadas. Cada uno de ellos se ha configurado con un color de fondo diferente para ayudar en el proceso de desarrollo y poder identificarlos fácilmente. La primera imagen muestra el navegador safari de escritorio, el cual tiene un fondo azul y con la siguiente muestra de diseño.

Figura 21. Aplicación vista desde el navegador Safari en el escritorio.



Figura 22. Aplicación en el navegador de Android.



Figura 23. Aplicación en el navegador de iPhone.



La hoja de estilo principal es la siguiente, llamada *netmon.js*:

Listado 2. Hoja de estilo primeria.

```
body {
  color: #888888;
  font-family: Helvetica;
  font-size:14px;
  margin: 0px;
  padding: 0;
}
.details {
  margin: 0px;
  padding: 0px;
  background-color: white;
  border: solid;
  border-width: 1px;

  -webkit-border-top-left-radius: 8px;
  -webkit-border-top-right-radius: 8px;
  -webkit-border-bottom-left-radius: 8px;
  -webkit-border-bottom-right-radius: 8px;
}
.OK {
  color: #000000;
}
.BAD {
  color: #ff0000;
}
.odd {
  background-image: -webkit-gradient(linear, left top, right bottom,from(#ccc)
,to(#999));
}
.even {
  background-image: -webkit-gradient(linear, left top, right bottom,from(#999)
,to(#ccc));
}
.serverentry a {
  float: right;
  color: #ffffff;
}
.serveritems{
  color: #000;
}

#header h1 {
  margin: 0;
  padding: 0;
  text-align: center;
  color: #000;
}
```

Las hojas de estilo específicas de la plataforma logran tres objetivos principales:

1. Ajustar el tamaño de fuente entre el escritorio y las plataformas móviles.
2. Cambiar la combinación de colores como demostración en bruto de lo que impacta una hoja de estilo y demostrar lo fácil que es apuntar a una hoja de estilo específica para una plataforma en particular sobre la base de la userAgent.
3. Para ejercer funciones específicas en WebKit. Esto es importante cuando nos dirigimos a un navegador que no soporte WebKit.

A modo de ejemplo, el archivo `iphone.css` que se muestra a continuación:

Listado 3. Archivo `iphone.css`

```
body {
  background-color: #00ff00;
}
.serverentry {
  -webkit-border-top-left-radius: 8px;
  -webkit-border-top-right-radius: 8px;
  -webkit-border-bottom-left-radius: 8px;
  -webkit-border-bottom-right-radius: 8px;
}
.name {
  font-size: 2em;
}
.summary{
  font-size: 1.5em;
}
.serverentry a {
  font-size: 1.5em;
}
```

En este archivo, se ve que el color de fondo del cuerpo de la etiqueta se establece en verde (# 00FF00) y algunos tamaños de fuente para hacer las entradas de lectura más fáciles en el dispositivo móvil.

Por último, vamos a echar un vistazo a `netmon.js`, el cual contiene una lista de entradas, junto con una función diseñada para hacer una entrada, se muestra en la siguiente imagen. Se muestra una pequeña parte del código:

Listado 4. `Netmon.js`

```
var netmon = {
  initialize : function () {
  },
  resources :
  [
    {
      name : 'msiservices.com',
      homeurl : 'http://msiservices.com',
      pingurl : 'http://msiservices.com/netmon.php',
      status : 'OK',
      summary : 'OK',
      items :
      [
        {name : 'DiskSpace', value : '22.13 GB'},
        {name : 'Database Up?', value : 'Yes'}
      ]
    },
    {
      name : 'server 2',
      homeurl : 'http://someurl',
      pingurl : 'http://someurl/netmon.jsp',
      status : 'OK',
      summary : 'OK',
      items :
      [
        {name : 'DiskSpace', value : '100.8 GB'},
        {name : 'Database Up?', value : 'Yes'}
      ]
    }
  ]
}
```

```
},
// additional entries clipped for brevity

],
render : function(index,itm) {
  try {
    var ret = "";
    ret += "<div class='serverentry " + itm.status + " " + (index % 2 == 0 ?
'even' : 'odd') + "'>";
    ret += "<span class='name'>" + itm.name +
"</span>&nbsp;&nbsp;&nbsp;<a target='_blank' href='" + itm.homeurl +
">Show</a><br />";
    if (itm.status != "OK") {
      ret += "<span class='summary'>-" + itm.summary +
"</span><br />";
    }

    ret += "<div class='serveritems'>";
    jQuery.each(itm.items,function (j,itemdetail) {
      ret += ">>" + itemdetail.name + "=" + itemdetail.value +
"<br />";
    });
    ret += "</div>";
    ret += "</div>";
    return ret;
  } catch (e) {
    return "<div class='error'>Error rendering item [" + itm.name + "]"
    + e + "</div>";
  }
}
};
```

Cuando una entrada del servidor no se encuentra OK, esta se muestra en rojo, gracias a que se ha definido mediante una clase en el archivo CSS. A demás, cuando el estado no es correcto, se muestra un campo de resumen para proporcionar de un solo vistazo el problema.

3.2.4.1 Tomar una aplicación web móvil como aplicación de escritorio

Solo tenemos que imaginar que se quiere vender o rentar una aplicación web de monitoreo. Sin duda se puede vender a través de una suscripción tradicional o modelo SaaS, pero si lo que se desea, es empaquetar la aplicación y venderla a través de un mercado de aplicaciones, como el App Store de iTunes o el Market de Google, con el fin de hacer eso, la aplicación debe ser compilada como una aplicación nativa. Afortunadamente, existen soluciones para esto.

Cada plataforma móvil importante, dispone de un medio para integrar un navegador en una vista o formulario, o actividad. Cada plataforma tiene una técnica diferente para hacer esto, pero todos ellos trabajan de una forma similar. Un control de explorador se coloca dentro de la aplicación nativa y esta puede interactuar con él. En el modelo más simple, el control del explorador puede simplemente acceder al contenido de su Web. Por otro lado, la aplicación puede interceptar solicitudes de enlace y proporcionar su propio contenido, lo que hace incrementar la vista de la presentación del navegador. Solo hay que recordar que HTML y CSS son alternativas compatibles con los widgets nativos, independientemente del origen del contenido de la solicitud. Algunas aplicaciones, serán un híbrido de los dos enfoques. Por ejemplo,

una aplicación puede obtener la mayoría de su contenido de la Web, pero sacar partido de su lado nativo, proporcionando acceso a recursos locales a través del GPS, bluetooth, incluso integrar el funcionamiento con otras aplicaciones nativas, etc.

Hay algunas herramientas disponibles en el mercado para crear una aplicación de escritorio con este tipo de arquitectura. PhoneGap y Appcelerator son dos ejemplos de las mismas.

3.3 Crear una aplicación basada en el navegador siendo multiplataforma

El uso de Ajax, jQuery, JavaScript, CSS Y HTML5

En la primera parte, se hace una introducción a WebKit, el motor del navegador en Android e iOS. En esta parte se va a profundizar, mediante la construcción de una aplicación de gestión de red que se ejecuta en la mayoría de los navegadores de dispositivos móviles más comunes. La aplicación muestra en el navegador local de almacenamiento de SQL y Ajax, tecnologías clave que permiten una rica y grata experiencia de aplicación desde el navegador del dispositivo móvil. Además, la aplicación emplea la popular librería jQuery de JavaScript.

3.3.1 Introducción

Volviendo la vista atrás en el tiempo, ha habido una serie de obstáculos para el desarrollo de aplicaciones Web móviles y dotarlas de funciones, pero actualmente esto ha cambiado. El desafío de la velocidad de la red se mejoró sustancialmente al surgir la tecnología 3G, y ahora aún mucho más con la velocidad 4G, que aún la mayoría de los dispositivos actuales no están preparados, pero cada vez más. Del mismo modo, en los comienzos la interfaz de usuario ha recorrido un largo camino, con el iPhone a la cabeza, y en Android ha ganado impulso con diseños innovadores, el potente motor WebKit HTML proporcionando lo mejor en su clase y renderizado de CSS. Aunque en todas las plataformas de dispositivos móviles, se puede observar cómo avanzan en este sentido, Windows Phone es otro claro ejemplo, todas ellas intentan dotar al usuario de una interfaz cada vez más intuitiva y fácil, pero a la vez más funcional.

Igual que la experiencia del explorador de escritorio, las aplicaciones Web móviles han experimentado una especie de renacimiento como el almacenamiento de base de datos local haciendo disponible para el navegador. La otra tecnología que encabeza aplicaciones Web móviles de última generación es Ajax. Esta tecnología es la encargada de utilizar JavaScript para invocar una página servidor para recuperar elementos de datos específicos a través de HTTP sin la necesidad de tener que recuperar y volver a representar toda la página Web. Ya que Ajax funciona de manera asíncrona, las aplicaciones Web móviles han ganado un gran impulso en la capacidad de función.

En esta parte del proyecto, se va a enfocar a iOS, Android y Windows Phone. Para ayudar en el desarrollo, también se va a probar la aplicación en Safari en el escritorio, debido a que Safari está basado en WebKit, es una plataforma de desarrollo ideal, en la que poder acelerar la presentación con la ayuda de una excelente depuración, gracias al uso de "WebKit Inspector application".

3.3.2 La aplicación

Antes de entrar en los detalles del código, se va a echar un vistazo a los objetos que emplea la aplicación. Si alguna vez se ha tenido un sitio Web, es probable que hayamos recibido una notificación de que este está caído. Puede venir de una herramienta de monitoreo, que sería lo ideal, o bien de un simple correo electrónico informando del mismo.

Cuando se abre el navegador y al acceder a la página de inicio vemos que está caído, puede estar relacionado con la conectividad, con lo cual, la primera opción que se debe tomar, antes de sumergirse para encontrar el problema, es volver a cargar el sitio Web.

Si se descarta una cuestión básica de cliente-conectividad, el siguiente paso es tratar de reunir algunos detalles clave, como la memoria disponible, los recursos del sistema de archivos, la memoria disponible, el último mensaje de error, etc. Para hacer esto, normalmente se requiere acceso al servidor a través de un escritorio remoto o una sesión de SHELL (SSH).

Recibir una notificación con este problema nunca es agradable, y más aún si en ese momento no se dispone de conectividad a Internet, lo que puede crear impotencia. Por ejemplo, si con el tiempo se ve que un sitio en particular tiende a fallar reportando el mismo tipo de errores. El problema puede derivar de un recurso externo, como puede ser una base de datos o un procesador de pago de terceros que no se encuentre disponible; o tal vez que una transferencia de archivos haya fallado. Sea cual sea la procedencia del fallo, hay algunas estadísticas claves que serían útiles a la hora de abordar inicialmente un problema. Estas estadísticas o indicadores de rendimiento, pueden variar de un sitio a otro. La construcción de una herramienta para ayudar a tomar este es el propósito de la aplicación que se construye como ejemplo.

La aplicación está diseñada para ayudar a tratar con problemas de sitio Web cuando se encuentra fuera del lugar de trabajo. Solamente con disponer de un dispositivo iOS o Android, se puede aprovechar las capacidades de estas potentes plataformas para ayudar a administrar los sitios Web.

3.3.3 Consideraciones de diseño

La mayor consideración de esta aplicación es que sea lo más independiente posible de una base de datos del servidor, lo que quiere decir, en otras palabras, que esta aplicación se ejecute sin la necesidad de crear una cuenta en un servicio de terceros. La aplicación que se va a crear para que el usuario solo tenga que descargar la página Web de una vez y ejecutarla desde el navegador de forma local. Lógicamente, el usuario podrá marcar y actualizar siempre que quiera para obtener las nuevas características que se hayan incluido en la aplicación en cualquier momento. Ya que todos los datos se almacenarán en el dispositivo móvil en una base de datos SQL local.

Aunque, sin poner en duda las ventajas de tener este tipo de datos almacenados en un servidor, la funcionalidad y almacenar los datos para múltiples usuarios está fuera del alcance de esta aplicación. La preocupación principal es aprovechar el almacenamiento

local y Ajax para construir una aplicación práctica y útil. A continuación, se exponen algunos pasos lógicos para la ampliación de la aplicación.

El código para esta aplicación se puede dividir en dos grupos distintos. Por un lado, nos encontramos con un código que se ejecuta en el dispositivo móvil, el cual incluye:

- Un archivo `index.html` que sería la Shell de la aplicación.
- Un archivo JavaScript llamado `netmon.js`, que contiene la mayoría de la funcionalidad de la aplicación.
- Otro archivo llamado `json2.js`, que contiene las rutinas relacionadas con JSON.
- Múltiples archivos CSS que contienen la implementación sobre los estilos. Hay que recordar, que como se vio anteriormente, tenemos un archivo CSS principal, que contiene la mayor parte de los estilos, y otros archivos CSS que contienen información específica, con un perfeccionamiento de la apariencia según la plataforma desde donde se acceda.
- Se incluye el archivo de biblioteca `jquery.js` para ayudar en la manipulación de DOM y consultas Ajax.

Por otro lado, tenemos el código que se ejecuta en el servidor y es único para cada sitio que queremos gestionar. Los detalles de este código pueden variar en cada aplicación, pero el contenido resultante es siempre la misma JSON (JavaScript Object Notation), que contiene una serie de propiedades específicas, elementos de propiedades con nombre (conjunto de pares de nombre y valor). Los datos dentro de este objeto JSON, marca la forma en que se mostrará la aplicación en el navegador. Además, estos pares de nombre/valor, dotarán de datos operativos clave específicos para cada sitio, proporcionando (o eso se espera) la información necesaria para clasificar la situación rápidamente y ayudar a la persona que tenga acceso al servidor para resolver el problema lo antes posible.

3.3.4 La construcción de la aplicación

En un principio, el estudio del modelo de datos de esta aplicación, una vez se tenga una idea del modelo de datos, se pasa a examinar el código que interactúa con los datos, con lo que la aplicación cobra vida.

3.3.4.1 Modelo de datos

Cuando los datos se almacenan en el dispositivo móvil a través del navegador, que se almacena en una base de datos de HTML 5, o una tienda de base de datos SQL/navegador accesible. La especificación de datos SQL basada en un navegador móvil, aún está en proceso de cambio. Sin embargo, en términos prácticos, hay disponible para usar en algunos dispositivos móviles y otros entornos WebKit habilitados. Esta funcionalidad de base de datos es esencialmente una interfaz de JavaScript para una implementación de SQLite subyacente. La base de datos contiene una sola tabla: `tbl_resources`.

La tabla de los datos de la aplicación utiliza en tiempo de ejecución, esencialmente es una matriz de objetos JSON. Cada objeto de la matriz contiene:

- Nombre del sitio.
- Página de inicio de la Web.
- URL de las estadísticas clave del sitio; a esto se le suele denominar *ping URL*.
- Estado actual del sitio: Bueno o malo.
- Resumen actual del sitio: es una breve descripción del estado actual del sitio. Esto podría parecer la base de datos en forma resumida, por ejemplo.
- Conjunto de pares de nombre/valor que contienen datos específicos del sitio para ayudar en la descripción de las condiciones actuales de operación del sitio. Hay que tener en cuenta, que estos valores pueden ser importantes, incluso cuando un sitio no está caído. Un elemento particular puede contener datos que da una visión a un problema inminente en un futuro próximo.

El Listado 5 es un ejemplo de objeto JSON que representa un sitio.

```
[
  {
    name : 'msi-wireless.com',
    homeurl : 'http://msiservices.com',
    pingurl : 'http://ibm.msi-wireless.com/mobile2/netmon.php',
    status : 'OK',
    summary : 'Everything is fine...',
    items :
    [
      {name : 'DiskSpace', value : '22.13 GB'},
      {name : 'Database Up?', value : 'Yes'}
    ]
  }
]
```

El papel de base de datos es para preservar, almacenar y utilizar los datos. La manipulación se realiza a través de una matriz que contiene estos objetos, en lugar de dirigirse a la base de datos continuamente. Esta estrategia se implementó para simplificar un poco la metodología en el código JavaScript y reducir al mínimo el número de veces que se accede a la base de datos para meter y sacar datos de la misma. Por otro lado, si la aplicación maneja un volumen grande de datos, puede ser positivo trabajar directamente con la base de datos. Existe también otra forma de trabajar si el número de datos es elevado, se le suele denominar paginación, donde se toma una serie de elementos de la base de datos denominada “ventana”, que representa un subconjunto del número total y una matriz JavaScript que se ejecuta sobre ella.

La Figura 24, contiene un resumen de la estructura de base de datos y unos cuantos registros como ejemplo. La visualización de la base de datos se puede lograr con el Inspector Web, que de este elemento, disponen la mayoría de los navegadores en la

actualidad, aunque en los inicios fue Safari / WebKit. Esta es una de las razones por las que al desarrollar con WebKit es ventajoso.

Figura 24. Captura de la estructura de la base de datos.

name	homeurl	pingurl	status	summary	items
msiser...	http://ms...	http://ibm.msi-wireless.co...	BAD	[the sky is f...	[{"name":"Date","value":"11/20/09 12:...
ibm d...	http://go...	http://ibm.msi-wireless.co...	OK	No worries...	[{"name":"weather","value":"sunny"}]
ibm d...	http://bi...	http://ibm.msi-wireless.co...	BAD	No more c...	[{"name":"coffee pot","value":"empty"}]

Ahora se puede tener una idea de los elementos de la base de datos. Por otro lado, el manejo para la aplicación.

3.3.5 El lado del cliente – La vista por defecto

La interfaz de usuario por defecto de la aplicación, es una lista de los sitios que se administran y ordenados por su estado. Los sitios que no se encuentran en su estado óptimo se encuentran en primer lugar, como se ve en la *Figura 2*.

Figura 25. Los sitios que no funcionan del todo bien



Según se puede observar, tenemos un par de sitios bajo gestión. Actualmente, uno de ellos nos reporta un problema. Si un sitio se encuentra en buenas condiciones, es decir, su revisión de estado es igual a OK, no se muestra el campo resumen, y se muestra el texto negro. Si el estado es desfavorable, se muestra el resumen al lado del nombre del sitio y un estilo llamado BAD en el archivo CSS marca los atributos de representación, en este caso el texto en rojo. Para más detalles, se puede consultar el archivo netmon.css.

Al hacer clic sobre una entrada, los detalles para alternar entre visible o no. Como se observa en la *Figura 2*, cada entrada tiene tres vínculos disponibles, seguido de los lugares de origen y de la URL de ping, y además una sección de una lista de los detalles de la página, es decir, la lista de pares de nombre/valor que representan las condiciones para ese sitio.

En este ejemplo, vemos que el sitio denominado “ibm demo 2” tiene el resumen de “no more coffee”, esto por supuesto no tiene por qué ser una urgencia técnica, sino que proporciona un ejemplo a considerar. Pasando a la sección de detalles de esta entrada, vemos la estadística clave detrás de esta condición del servidor: “The Coffee Pot is empty”.

Vemos que podemos pinchar sobre el vínculo **Home**, que pondrá en marcha una nueva ventana del navegador. En segundo lugar, podemos actualizar los datos pulsando sobre el enlace **Refresh**.

Por último, podemos quitar esta entrada mediante la sección del enlace **Remove**. Una simple consulta `widow.confirm()` nos pide que se confirme si se quiere realizar esta acción de manera no reversible.

3.3.6 El lado del cliente – El HTML

Para crear esta lista, vamos a entrar en detalle en dos archivos. El primero es `index.html`, que se muestra en el Listado 2, y el segundo es `netmon.js`, que se muestra en el Listado 3. Dentro de este proyecto se explican varios fragmentos del código, aunque se adjunta al completo.

Todavía, el punto principal, sigue siendo el WebKit móvil meta viewport para ayudar a guiar al navegador cómo preferimos la presentación de la página. También se usa algo de JavaScript selectivo para especificar un archivo CSS en particular y así poder diversificar según el dispositivo.

Lo que sí se puede considerar novedoso es la inclusión de algunos archivos JS junto con algunos archivos JavaScript “locales”. El archivo se completa con un nuevo `HTML div` con un ID `entryform`, que contiene los elementos necesarios para que el usuario pueda agregar perfectamente una nueva entrada desde la aplicación. No es necesario tener que cargar una nueva página HTML, que es lo que se suele realizar comúnmente para esta tarea. Recordemos que uno de los objetivos de diseño es que la aplicación no tenga que basarse en las herramientas basadas en servidores adicionales para poder modificar los datos o almacenarlos desde el exterior de los sitios que monitoreamos. El Listado 2 incluye las declaraciones de `index.html`.

Listado 6. Index.html

```
<link rel="stylesheet" href="netmon.css" type="text/css" />
<script type="text/javascript" src="jquery.js"> </ script>
<script type="text/javascript" src="netmon.js"> </ script>
<script type="text/javascript" src="json2.js"> </ script>

<script type="text/javascript">
  if (navigator.userAgent.indexOf ('iPhone')! = -1) {
    document.write ('<link rel="stylesheet" href="iphone.css" type="text/css" />');
  } Else if (navigator.userAgent.indexOf ('Android')! = -1) {
    document.write ('<link rel="stylesheet" href="android.css" type="text/css" />');
  } Else {
    document.write ('<link rel="stylesheet" href="desktop.css" type="text/css" />');
  }
}
```

3.3.7 El lado del cliente – Rellenar la lista

Los datos se extraen de nuestra base de datos local y se muestran en una lista de entradas de servidor. Para poder obtener esta lista, se abre la base de datos y se emite una consulta SQL para obtener todos los elementos dentro de nuestra tabla base.

Una técnica común en las aplicaciones de este tipo es la creación de tablas dinámicas sobre la marcha. Hay que recordar, que para esta aplicación no se dispone de un DBA alrededor para ayudar a realizar estas tareas, la aplicación tiene que realizar todo el trabajo por cuenta propia. Dentro del archivo *netmon.js* se implementan una serie de elementos, incluyendo:

- Definición de un constructor de tipo de datos para el objeto *netmonResource*.
- Un grupo de funciones necesarias para ayudar a la aplicación en la interacción con la base de datos y los sitios Web que van a ser monitoreados. Todo esto se desarrolla en el espacio de nombres *netmon* y contiene cosas como las siguientes:
 - *init* – Una función utilizada para crear / abrir la base de datos y poder recuperar la lista de servidores que se tiene. *Listado 3* contiene esta función, que intenta abrir la base de datos (si ya no está abierta) y consultar la misma. Si no tenemos la tabla en la base de datos, esta función inicia el proceso de creación.
 - *dbHandle* – Una función utilizada para gestionar la conexión con la base de datos.
 - *addEntry* – Función para insertar registros en la base de datos.
 - *createResourcesTable* – Función que crea la tabla necesaria en la base de datos con un solo registro por defecto de demostración.
 - *refreshEntry* – Función para refrescar la entrada mediante la recuperación de ping a la URL mediante una llamada a Ajax.
 - *DeleteEntry* – Función para eliminar una entrada de la tabla de la base de datos.
 - *Render* – Función que devuelve una cadena HTML, con el formato que se idea mediante algunas reglas sobre como preferimos cada entrada de los servidores que se muestran. Si surgen nuevas ideas de formato, se llevarían a cabo con esta función, juntos con los estilos CSS.
 - *Resources* – Matriz que contiene la caché de las entradas de la base de datos. Son algunas de las otras funciones de trabajo en esta serie, en lugar de la base de datos directamente.

Listado 7. *Netmon.js*

```
initialize : function () {  
    try {
```

```

if (netmon.dbHandle == null) {
    netmon.dbHandle = openDatabase("netmondb","1.0","netmon",1000);
}

$("#mainContent").html("");
netmon.dbHandle.transaction(function(tx) {
    tx.executeSql("SELECT * FROM tbl_resources order by status,nme",
        // parameters
        [],
        // good result
        function(tx,result){
            try {
                var items = new Array();
                for (var i = 0; i < result.rows.length; i++) {
                    var row = result.rows.item(i);
                    items[i] = new
netmonResource();

                    items[i].name = row['nme'];
                    items[i].homeurl =
row['homeurl'];

                    items[i].pingurl =
row['pingurl'];

                    items[i].status = row['status'];
                    items[i].summary =
row['summary'];

                    items[i].items =
eval(row['items']);

                    if (items[i].items ==
undefined) {

                        items[i].items = [];
                    }
                }
                netmon.resources = items.slice(0);

                // setup gui with our data
                if (netmon.resources.length > 0) {
                    jQuery.each(netmon.resources,function (index, value) {
                        $("#mainContent").append(netmon.render(index,value));
                    });
                    $(".serverentry").click (function() {$(this).find(".serveritems").toggle();});
                    $(".serveritems").hide();
                }
            } catch (e) {
                alert("Error fetching rows from database..." + e);
            }
        },
        function(tx,error){
            //alert("bad result on query: " + error.message + "
            let's try to create the db table");
            netmon.createResourcesTable();
        }
    );
});
} catch (e) {
    alert("error opening database. " + e);
}
},
...
render : function(index,itm) {
    try {

```

```

var ret = "";
ret += "<div class='serverentry ' + itm.status + " " + (index % 2 == 0 ? 'even' : 'odd') + "'>";
//ret += "<span class='name'>" + itm.name +
"</span>&nbsp;&nbsp;&nbsp;<a target='_blank' href='" +
itm.homeurl + "'>Show</a><br /><a target='_blank'
href='javascript:netmon.deleteEntry(\"" + itm.name + "\");'>Remove</a><br />";
ret += "<span class='name'>" + itm.name + "</span>";
if (itm.status != "OK") {
    ret += "<span class='summary'>-" + itm.summary + "</span><br />";
}

ret += "<div class='serveritems'>";
ret += "<a target='_blank' href='" + itm.homeurl + "'>Home</a>&nbsp;&nbsp;&nbsp;";
<a target='_blank' href='javascript:netmon.refreshEntry(" + index + ",false);'>Refresh</a>&nbsp;&nbsp;&nbsp;";
<a target='_blank' href='javascript:netmon.deleteEntry(\"" + itm.name + "\");'>Remove</a><br />";
ret += "Home URL:&nbsp;&nbsp;&nbsp;" + itm.homeurl + "<br />";
ret += "PING URL:&nbsp;&nbsp;&nbsp;" + itm.pingurl + "<br />";
ret += "<hr />Details<br />";
jQuery.each(itm.items,function (j,itemdetail) {
    ret += ">>" + itemdetail.name + "=" + itemdetail.value + "<br />";
});
ret += "</div>";
ret += "</div>";
return ret;
} catch (e) {
    return "<div class='error'>Error rendering item [" + itm.name + "]" + e + "</div>";
}
}
}

```

El espacio de nombres netmon contiene la mayoría de la lógica de lo que se necesita para implementar el sitio, complementado por algunas funciones auxiliares en el archivo *index.html*. A continuación, se va a echar un vistazo a la tarea de agregar una nueva entrada a la base de datos.

3.3.7.1 El lado del cliente – Agregación de un sitio

La aplicación, como ya se ha dicho, gestiona una lista de uno o más sitios de interés para el usuario. Para cada sitio, inicialmente se recoge y guarda:

- El nombre del sitio.
- Home URL.
- Ping URL.

Para lograr esto, se crea de una forma simple, un frame que contiene tres campos, sus etiquetas, y un par de botones, como se muestra en la *Figura 26*.

Figura 26. Forma simple.



The image shows a web form titled "My Servers" on a blue background. At the top, there are two links: "Add New Server" and "Refresh Status". Below these are three input fields: "Site Name" with the value "my site name", "Home URL" with the value "http://myserverurl", and "Ping URL" with the value "http://myserverurl/stats.php". At the bottom of the form are two buttons: "Save" and "Cancel".

Este contenido del formulario se encuentra dentro de la misma página *index.html*, como se muestra en el *Listado 6*. Se cambia entre la vista de lista predeterminada y esta forma de añadir un nuevo servidor por medio de jQuery. La lista de entradas del sitio está dentro de un HTML *div* con un ID de *mainContent*. La forma está dentro de un HTML *div* con ID de *entryform*, que inicialmente se encuentra oculto. En cualquier momento, la visibilidad de estos *div* se excluir. Para cambiar la visibilidad, se llama al método *alternate* jQuery, en cada uno de estos elementos, como se muestra en la función *addNewEntry*, que se muestra en el *Listado 4*.

Listado 8. La función *addNewEntry*

```
función addNewEntry () {  
  . $ ("# Nombre_entrada") val ("");  
  . $ ("# Entry_homeurl") val ("");  
  . $ ("# Entry_pingurl") val ("");  
  
  . $ ("# MainContent") toggle ();  
  . $ ("# Entryform") toggle ();  
}
```

Cuando se ha hecho una nueva entrada de un servidor, se tiene que guardar en la base de datos y actualizar la lista de servidores para reflejar la inclusión de la nueva entrada en la lista. El usuario inicia seleccionando el botón **Guardar**, que invoca la función *saveEntry* de JavaScript implementada en el archivo *index.html*. Esta función lo que hace es crear un nuevo objeto de JavaScript llamado *netmonResource*. Se valida que todos los campos se han cumplimentado correctamente, y a continuación, se inicia el ahorro de la base de datos. Se guarda la nueva entrada invocando *netmon.addEntry*. Después, se lanza de nuevo la lista de entradas para llamar otra vez a las dos entradas primarias de los elementos "div". Nota: Se implementa la expresión regular para validar las direcciones URL, lo que sería una buena idea.

Para la siguiente explicación, consultar la Ficha 3: *netmon.sj*.

Mediante la interfaz de base de datos SQL en JavaScript requiere dos pasos básicos. En primer lugar, hay que abrir la base de datos si no se ha abierto previamente. En esta aplicación, se abre la base de datos en la función de inicialización. Para crear un nuevo registro en la base de datos, se va a examinar el código en la función *addEntry*, que se muestra en el *Listado 9* como se inicia llamando a una transacción SQL.

<databasehandle> transacción (function (tx) {tx.executeSql ({})}); . En este caso, la raíz de la base de datos es *netmon.dbHandle*.

La función *ExecuteSQL* toma cuatro argumentos, que son los siguientes:

1. Una sentencia SQL en la que los parámetros, tales como valores de campo, se representan con un marcador de posición de ?. Cada ? Se sustituye con un valor, basado en la posición ordinal que figura en el segundo argumento de la función.
2. Una matriz con los valores utilizados como parámetros, sustituyendo a los ? Marcadores de posición como se indica en la instrucción SQL en el primer argumento.
3. Una función que devuelve la llamada e invoca si la sentencia se ejecuta con éxito. Los argumentos de esta función de devolución de llamada incluyen un grupo de transacciones y un conjunto de resultados.
4. Una función de devolución de llamada invoca si la declaración ha detectado algún error. Los argumentos de esta función de devolución de llamada incluyen un grupo de transacciones y un objeto de error.

El listado 9. La función *netmon.addentry*

```
addEntry : function (entry) {
  try {
    netmon.dbHandle.transaction(function(tx) {
      tx.executeSql("insert into tbl_resources (nme,homeurl,pingurl,status,summary,items) values (?,?,?,?,?,?)",
        [
          entry.name,
          entry.homeurl,
          entry.pingurl,
          entry.status,
          entry.summary,
          JSON.stringify(entry.items)
        ],
        function (tx,results) {
          // alert(entry.name + " added.");
          netmon.initialize();
        },
        function (tx,error) {
          alert("Error in addEntry [" + error.message + "]");
        });
    });
  } catch (e) {
    alert("Error in netmon.addEntry " + e);
  }
}
```

Se ha visto ya la posibilidad de agregar entradas, para la eliminación de entradas de la lista se hace de forma muy similar. Solo hay que utilizar la función *netmon.deleteEntry* para eliminar entradas de la base de datos.

Ahora hay que ver como se refrescan entradas en Ajax.

3.3.8 El lado del cliente – Actualizar con Ajax

Anteriormente, como bien se explicó, Ajax es una técnica para recuperar contenido del servidor sin necesidad de tener que actualizar toda la página Web. Esto se hace posible entremezclando diferentes tecnologías subyacentes en función del navegador. Sin embargo, se ha tomado el enfoque utilizando la biblioteca jQuery JavaScript para ello. Esto está enfocado para desarrollar en la mayoría de navegadores de dispositivos móviles existentes.

La función *netmon.refreshEntry* contiene el código en Ajax, que se muestra en el Listado 10, así como algo de código para actualizar la base de datos con los datos resultantes.

Listado 10. La función *netmon.refreshEntry*

```
refreshEntry : function (entryidx,bfollow) {
  try {
    //alert("refresEntry [" + netmon.resources[entryidx].name + "]" + netmon.resources[entryidx].pingurl +
    "");
    $.get(netmon.resources[entryidx].pingurl,
    function (data,textstatus) {
      // alert("response is here : [" + data + "]");
      try {
        var response = eval(data);

        netmon.dbHandle.transaction(function(tx) {
          tx.executeSql("update tbl_resources set status = ?,summary = ?,items = ? where nme = ?",
          [
            response[0].status,
            response[0].summary,
            JSON.stringify(response[0].items),
            netmon.resources[entryidx].name
          ],
          function (tx,results) {
            netmon.initialize();
            if (bfollow) {
              if (entryidx + 1 < netmon.resources.length) {
                netmon.refreshEntry(entryidx + 1,true);
              }
            }
          },
          function (tx,error) {
            //alert("Error in refreshEntry [" + error.message + "]");
            if (bfollow) {
              if (entryidx + 1 < netmon.resources.length) {
                netmon.refreshEntry(entryidx + 1,true);
              }
            }
          }
        ));
      }
    });
  } catch (e) {
    alert("error handling response [" + e + "]");
  }
}

} catch (e) {
  alert("Error refreshEntry " + e);
}
}
```

Para obtener datos del servidor, simplemente se invoca “\$.get()”. El primer argumento es la dirección URL del sitio, y el segundo es una función que se invoca cuando se completa la llamada. En el ejemplo, se actualiza la base de datos con los datos más recientes recuperados (ver Listado 3: *netmon.refreshEntry*) mediante el uso de algunas de las habilidades de la base de datos ya expuestos anteriormente.

jQuery también prevé la posibilidad de proporcionar “los datos del formulario en código” para pasarlos por parámetros a una página Web, y a la vez, da una pista el tipo de datos que se esperan de nuevo. Ver “Adjunto: **os-androidiphone2-browserwars2.zip**” para tener una referencia a las funciones de Ajax de jQuery.

Hay que tener en cuenta, que jQuery también posee un método para recuperar un objeto JSON directamente. Sin embargo, se optó por la función genérica de Ajax.

Cuando se reciben los datos desde el servidor, se quiere convertir en un objeto de JavaScript para acceder a sus propiedades. Pudiendo hacer esto de forma sencilla con la función de JavaScript llamada: `eval : objet var = eval (<text <json);`

3.3.9 El lado del servidor – Construir tu propia página para monitorear

Se pasa a describir la estructura JSON en nuestra aplicación. Para poder construir de penderá de las tecnologías con las que se quiera trabajar, por ejemplo si se prefiere con .NET, se puede generar una página aspx, para generar dinámicamente los datos. De forma alternativa, se puede desarrollar un script de SHELL, programado periódicamente para que genere un archivo de texto y así poder recuperarlo y averiguar el estado de la última actualización. La conclusión es que la aplicación debe tener la capacidad de recoger un objeto JSON basado en una única URL, pero esta función se deja a parte.

Listado 11. Archivos PHP de muestra

```
<?
$filecount = 0;
$statusValue = "OK";
$summaryValue = "No Error";
?>
[
{
items : [
{"name" : "Date", "value" : "<? echo date("m/d/y H:m:s"); ?>" },
<?
foreach (new DirectoryIterator("./data") as $fileInfo) {
if ($fileInfo->isDot() || $fileInfo->isDir()) continue;
$filecount++;
echo "{ \"name\" : \"\".$fileInfo->getFilename().\"\", \"value\" :
\"\".$fileInfo->getSize().\"\", \"";
if ($fileInfo->getFilename() == "error.txt") {
if ($fileInfo->getSize() > 0) {
$statusValue = "BAD";
$size = $fileInfo->getSize();
$fh = fopen("./data/".$fileInfo->getFilename(),"r");
$summaryValue = fread($fh,$size);
fclose($fh);
}
}
}
}
?>
{"name" : "FileCount", "value" : "<? echo $filecount ?>"}
}
```

```

],
"status": "<? echo $statusValue ?>",
"summary": "<? print str_replace("\n","",$summaryValue) ?>"
}
]

```

Este código genera los siguientes datos JSON que se muestran a continuación en el Listado 12, contemplando incluso una condición de error.

Listado 12. Datos JSON generados por el Listado 8

```

[
  {
    "items": [
      {
        "name": "Date",
        "value": "11/22/09 15:11:51"
      },
      {
        "name": "error.txt",
        "value": "20"
      },
      {
        "name": "bad.html",
        "value": "91"
      },
      {
        "name": "testfile.txt",
        "value": "44"
      },
      {
        "name": "good.html",
        "value": "87"
      },
      {
        "name": "FileCount",
        "value": "4"
      }
    ],
    "status": "BAD",
    "summary": "the sky is falling."
  }
]

```

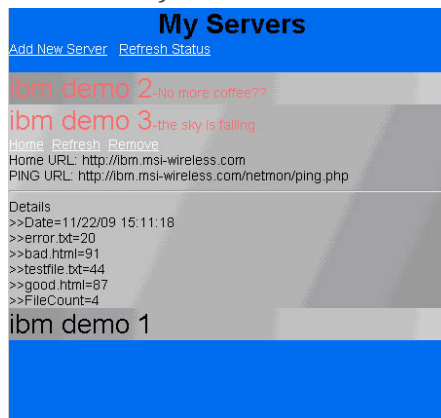
En este ejemplo, se proporciona la fecha en la que se generan los datos, una lista de los archivos y sus tamaños, así como un recuento actual de estos. Hay que tener en cuenta que la zona horaria del servidor puede ser diferente de la ubicación donde nos encontremos.

Para controlar y gestionar los errores que surjan, si *error.txt* no es de cero bytes de longitud, se lee el contenido de ese archivo y se le asigna a la propiedad de resumen. Es únicamente un script básico de monitoreo.

Mientras se trabaja en la generación del JSON, se puede encontrar un problema al analizar el código desde la aplicación basada en el navegador. Si este es el caso, es posible que se desee utilizar una herramienta de validación JSON, ver “Adjunto: os-

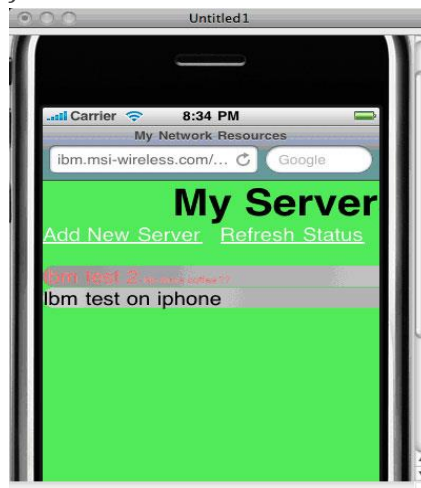
androidiphone2-browserwars2.zip". En la *Figura 4*, vemos como muestra esta entrada dictada en el navegador de escritorio.

Figura 27. Herramienta de validación de JSON



La Figura 28 muestra cómo se ejecuta la aplicación en un iPhone

Figura 28. Aplicación que se ejecuta en iPhone



La Figura 29 muestra la aplicación que se ejecuta en Android con slightly con diferentes listas de servidores.

Figura 29. Aplicación que se ejecuta en Android



La aplicación ya está al completo (más o menos). Siempre se pueden añadir cosas nuevas, o cambiar algunas, por lo que en la siguiente sección final, se va a ver una lista de cosas que se deja como exposición para ampliar.

3.3.9.1 Los próximos pasos

No hay escasez de ideas nuevas que pueden ser pensadas para mejorar y optimizar la aplicación. A continuación, se exponen algunos apuntes para mejorar:

- Implementar algunas expresiones regulares para validar correctamente los campos de entrada del formulario.
- Editar las entradas. Ahora mismo, solo se pueden añadir o eliminar.
- Adaptar a diferentes perfiles de empresa en el servidor para que se pueda pasar de un dispositivo a otro.
- Convertir esta aplicación en una aplicación nativa con alguna herramienta como “PhoneGap” o “Appcelerator”. Esto permitirá vender la aplicación o sacar rentabilidad de alguna forma con ella, en las tiendas de los dispositivos (diferente según el Sistema Operativo).
- Enviar información a un amigo por correo electrónico directamente desde la aplicación.

3.4 Magento mobile

Magento nos proporciona una extensión gratuita para añadirla al backend y poder convertir la tienda en un m-commerce. La instalación se hace mediante Magento Connect de la siguiente manera:

Figura 30. Instalación Extensión Magento Mobile.

Install New Extensions

- 1 Search for modules via [Magento Connect](#).
- 2 Paste extension key to install: **Install**

Extension dependencies

Channel	Package Name	Version	Status
community	Magento_Mobile	1.7.0.0.22.1 (stable)	Already installed
community	Mage_Core_Modules	1.7.0.0 (stable)	Already installed
community	Lib_Varien	1.7.0.0 (stable)	Already installed
community	Lib_Google_Checkout	1.5.0.0 (stable)	Already installed
community	Lib_Js_Calendar	1.51.1.1 (stable)	Already installed
community	Lib_Js_Mage	1.7.0.0 (stable)	Already installed
community	Lib_Phpseclib	1.5.0.0 (stable)	Already installed
community	Mage_Locale_en_US	1.7.0.0 (stable)	Already installed
community	Lib_Mage	1.7.0.0 (stable)	Already installed
community	Lib_ZF	1.11.1.0 (stable)	Already installed
community	Lib_Js_Prototype	1.7.0.0.3 (stable)	Already installed
community	Lib_ZF_Locale	1.11.1.0 (stable)	Already installed

Cancel installation **Proceed**

Al clicar en el botón “Proceed” la instalación comenzará, dando un resultado satisfactorio en el log:

Figura 31. Instalación Extensión Magento Mobile 2.

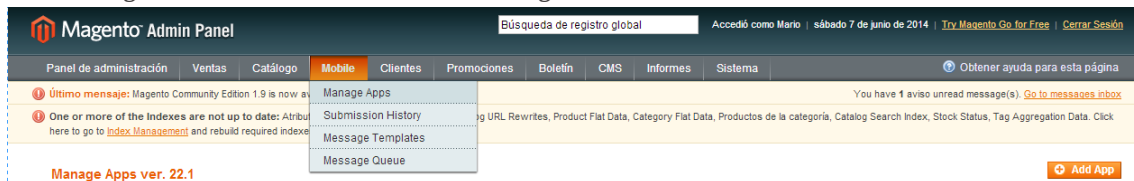
```
 Auto-scroll console contents

Checking dependencies of packages
Already installed: community/Magento_Mobile 1.7.0.0.22.1, skipping
Already installed: community/Mage_Core_Modules 1.7.0.0, skipping
Already installed: community/Lib_Varien 1.7.0.0, skipping
Already installed: community/Lib_Google_Checkout 1.5.0.0, skipping
Already installed: community/Lib_Js_Calendar 1.51.1.1, skipping
Already installed: community/Lib_Js_Mage 1.7.0.0, skipping
Already installed: community/Lib_Phpseclib 1.5.0.0, skipping
Already installed: community/Mage_Locale_en_US 1.7.0.0, skipping
Already installed: community/Lib_Mage 1.7.0.0, skipping
Already installed: community/Lib_ZF 1.11.1.0, skipping
Already installed: community/Lib_Js_Prototype 1.7.0.0.3, skipping
Already installed: community/Lib_ZF_Locale 1.11.1.0, skipping

Cleaning cache
***
Cache cleaned successfully
```

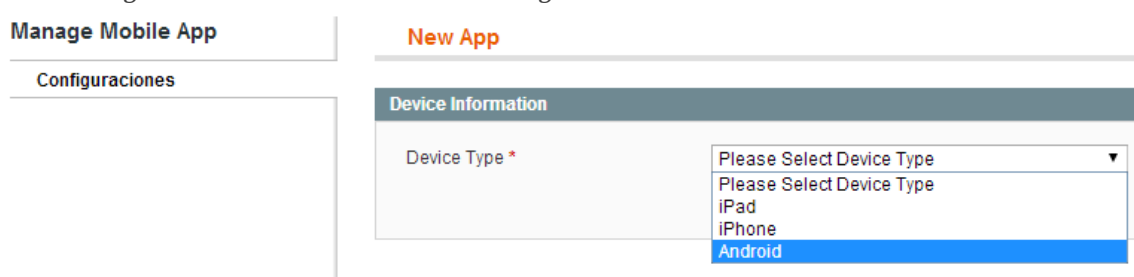
Para configurar la extensión en el backend accedemos a la siguiente parte del menú:

Figura 32. Instalación Extensión Magento Mobile 3.



Clicando en “Manage Apps” y luego en “Add App” para agregar la nueva aplicación móvil para la tienda.

Figura 33. Instalación Extensión Magento Mobile 4.



Estos son los tres tipos de dispositivos que se pueden elegir para configurar la herramienta Magento Mobile. La opción que voy a implementar va dirigida a los dispositivos que emplean Android como sistema operativo.

Ahora pasaremos a configurar cada uno de los módulos que nos proporciona la extensión para poder dar forma a la tienda y poder adaptarla lo mejor posible a lo que el cliente busca, dándole el diseño que se considere apropiado. A continuación, algunas imágenes que muestran la configuración que hemos ido modelando:

Figura 34. Instalación Extensión Magento Mobile 5.

En la pestaña de diseño:

Figura 35. Instalación Extensión Magento Mobile 6.

Una vez que se ha configurado todos los parámetros de la extensión, para hacer efectiva la aplicación y que este del todo activa para la tienda, hay que hacer “Submit” de la aplicación:

Figura 36. Instalación Extensión Magento Mobile 7.

Con pulsar en este botón la App aparecerá ya como activada para que funcione a través de navegador de internet, dando la siguiente información a rellenar por si queremos activarla para alguno de los Stores que hay, solicitando en el primer campo la clave de activación una vez se haya pagado:

Figura 37. Instalación Extensión Magento Mobile 8.



En el caso de que se quisiera subir la aplicación alguno de los Stores que poseen los diferentes dispositivos, en este caso sería para añadir al Google Play Sotore de Android, así los usuarios podrían acceder a la tienda con tan solo descargar e instalar la aplicación de Óptica Zaera en sus dispositivos con plataformas Android.

El único gran inconveniente es que si se quiere dotar a la tienda de esta última utilidad expuesta, es que hay que pagar un precio bastante alto. La aplicación nativa tiene un precio por cada dispositivo, y está disponible solo para los siguientes:

- Dispositivos iPhone.
- Plataformas Android.
- Dispositivos iPad.

El precio para cualquiera de los tres tipos de dispositivos es de 699€ en cada uno de ellos, e independientes, es decir, en el caso hipotético de que quisiéramos hacer extensible la aplicación nativa para que funcione en los tres tipos de dispositivos, habría que abonar:

$$699€ \times 3 \text{ dispositivos} = 2097 \text{ € al año}$$

Es una cantidad que creo se sale fuera de lo que realmente aportaría a la tienda. Salvo en algún caso en el que se quiera tener total compatibilidad, y lo más importante, dotar al backend de Magento teniendo integrado absolutamente todo y poder trabajar desde ahí todas las herramientas, en este caso, tanto la vista clásica (la que estaría orientada para los usuarios que accedan desde un ordenador), como la vista de la App para dispositivos móviles.

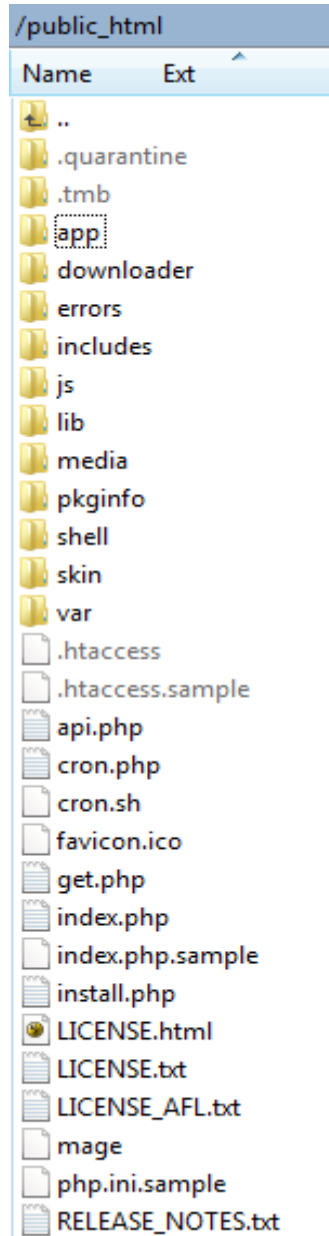
3.5 Desarrollo/Implementación de la Aplicación web para la tienda

Primero se va a explicar un poco como es la estructura de directorios (FILE SYSTEM), y más en detalle para los que afectan a la aplicación:

3.5.1 FILE SYSTEM

La estructura de directorios general es la siguiente:



Figura 38. Estructura de carpetas general de Magento.



Lo que contiene cada uno:

- ❑ App: Esta carpeta contiene los archivos de módulos, temas, de configuración y de traducción. También existen los archivos de plantillas para el “administrationtheme” por defecto y la instalación.
- ❑ Downloader: Esta carpeta es de almacenamiento de los archivos Web descargados. Se utilizan para la instalación y actualización de Magento a través de tu navegador.

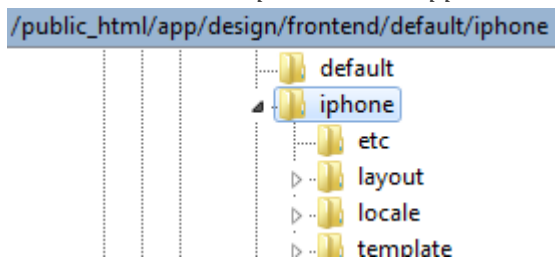
- ❏ Errors: Donde se almacena la funcionalidad y diseño de los errores que se mostrarán si la tienda no funciona correctamente.
- ❏ Includes: Archivos complementarios de configuración.
- ❏ Js: Contiene las librerías pre-compiladas de código JavaScript incluido en Magento.
- ❏ Lib: El código del núcleo de Magento se encuentra en esta carpeta. Contiene las bibliotecas de PHP del software.
- ❏ Media: Aquí se incluyen los archivos de medios de Magento, imágenes, miniaturas, carga las imágenes de los productos.
- ❏ Pkginfo: Contiene los archivos con la información sobre los cambios de los módulos de mejoras.
- ❏ Shell: Esta carpeta se genera si desde el backend de magento, cuando se vuelve a indexar los diferentes componentes.
- ❏ Skin: Aquí se localizan los archivos de los temas – Imágenes, JavaScript, archivos CSS, archivos Flash. También se pueden encontrar los archivos de para la instalación de skins y plantillas de la administración.
- ❏ Var: Caché, sesiones, copias de seguridad de bases de datos, exportación de datos y los informes de errores en caché se pueden encontrar en este directorio.
- 📄 .htaccess: Contiene reglas mod_rewrite, que son esenciales para las URLs y el motor de búsqueda. Aquí también se pueden encontrar directivas de servidor web y estándares PHP que pueden mejorar el rendimiento del sitio web.
- 📄 .htaccess.sample: se trata de una copia de seguridad del archivo htaccess. Si modifica htaccess que puede ser utilizado con el fin de obtener la configuración predeterminada.
- 📄 Cron.php: Un Cron Job se debe establecer para este archivo. La ejecución del archivo en un período de tiempo definido se asegurará de que el complicado sistema de almacenamiento en caché de Magento no afectará el rendimiento del sitio web.
- 📄 Favicon.ico: El favicon por defecto de Magento. Un pequeño icono que se muestra en la barra de herramientas del navegador una vez que el sitio web se carga.
- 📄 Index.php: El archivo principal de índice de Magento.
- 📄 Idex.php.sample: Una copia de seguridad del archivo de índice predeterminado. Se puede utilizar para revertir los cambios en un caso de una modificación index.php.
- 📄 LICENSE_AFL.txt: La licencia académica gratuita sobre la que se distribuye el software de Magento.
- 📄 LICENSE.txt: La licencia de Software Libre con la que se distribuye Magento.

-  Mage: Este archivo es el encargado de controlar la actualización automática, a través de la secuencia de comandos de descarga y SSH. Se ocupa de la actualización de cada módulo Magento individualmente.
-  Php.ini.sample: Este archivo contiene directivas de muestra php que se pueden utilizar con el fin de modificar la configuración de PHP. Si se desea modificar dicha modificación por defecto, se editará el archivo y se suprimirá “.sample” quedando como php.ini.

3.5.1.1 Directorio “app”

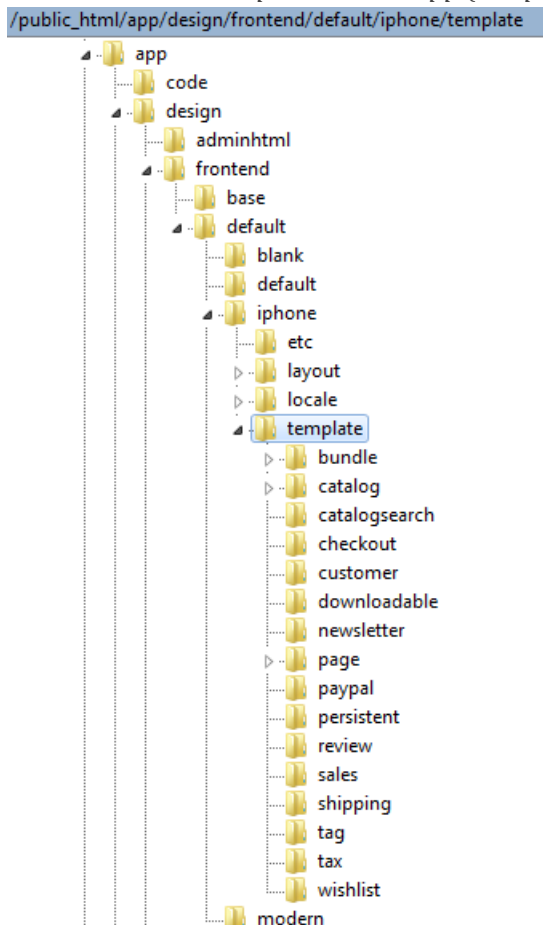
Dentro del siguiente directorio se encuentran los archivos de módulos, temas, de configuración y de traducción de la app:

Figura 39. Estructura de carpetas diseño app.



Dentro de template tenemos dividido por subcarpetas los diferentes módulos de la app:

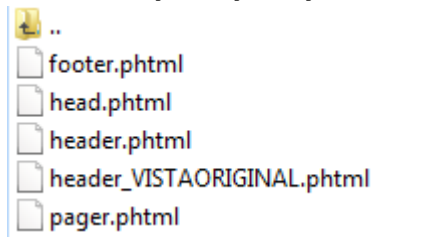
Figura 40. Estructura de carpetas diseño app (template).



En la siguiente carpeta se encuentran los principales bloques html:

/public_html/app/design/frontend/default/iphone/template/page/html

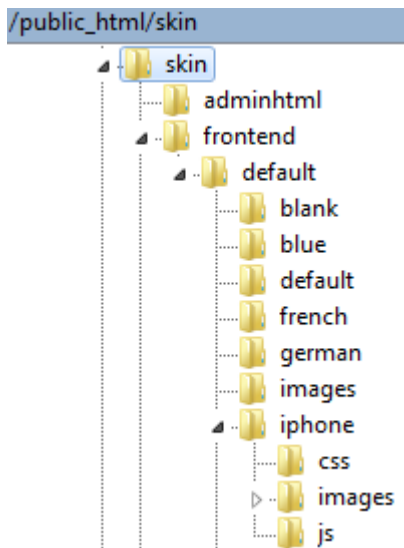
Figura 41. Archivos phtml principales de la app.



3.5.1.2 Directorio "skin"

Aquí están los archivos que definen el formato y diseño de la aplicación, JavaScript, CSS, imágenes, etc.

Figura 42. Estructura de carpetas diseño app (skin).



3.5.2 Excepción para funcionamiento de la Aplicación web:

Para que la Aplicación tenga efecto, y distinga si el usuario que está accediendo a la tienda, se puede configurar otro de los módulos que Magento provee desde el backend de la tienda.

Este módulo sirve también para aplicar diferentes diseños (Temas/Themes), para cambiar el aspecto de la tienda. Se pueden instalar temas predeterminados descargando e instalando desde Magento Connect.

Después, a la hora de subir productos, determinadas categorías, etc., podemos darle a cada uno de ellos el diseño que se quiera.

En este caso, se quiere añadir la excepción, para que según el dispositivo desde el que acceda el usuario:

- Si se intenta acceder desde un ordenador (PC), la vista de la tienda tendrá el diseño clásico, el que se ha visto hasta ahora.
- Si se intenta acceder desde un dispositivo móvil, Android o iPhone. Estas son las dos plataformas que Magento Mobile indica que son compatibles.

Para la configuración de la excepción, accediendo desde el backend en el menú: Sistema/Configuración, y desde la pestaña General/Diseño. La configuración habría que hacerla de la siguiente manera:

Figura 43. Creación de la excepción.

Por defecto se escoge el tema “default”, ya que si es un navegador que procede de un PC, la tienda se mostrará como la visión clásica, pudiendo aplicar cualquier otro tema como los que se han comentado anteriormente.

Por otro lado, para el resto de opciones, la excepción queda configurada como el tema iphone, ya que este tema aplica a la Aplicación creada con Magento Mobile. Y en el Campo “Matched Expresion” introducimos la concatenación de diferentes navegadores con los que se puede encontrar la tienda, a la hora de que un usuario intente acceder desde dispositivos montados en diferentes plataformas, e incluso, diferenciando entre diferentes modelos de una misma plataforma, como pueden ser las de la siguiente expresión:

iPhone|iPod|BlackBerry|Palm|Googlebot-Mobile|Mobile|mobile|mobi|Windows Phone|Windows Mobile|Safari Mobile|Android|Opera Mini

Después de guardar la expresión anteriormente creada, y de hacer un “flush” de las diferentes cachés, se prueba el acceso a la tienda desde diferentes dispositivos.

Finalmente, lo que se observa es que al acceder a la tienda, magento es capaz de diferenciar entre un dispositivo u otro cargando un diseño si el dispositivo pertenece a alguno de los mencionados en la expresión y el diseño por defecto si el dispositivo no se encuadra en ninguno de los de la excepción.

3.5.3 Activación del módulo Desarrollador Magento

Magento nos da la opción de configurarlo de tal forma que se pueda llevar un seguimiento del origen de los elementos que se cargan, ya que todo está bastante modularizado. Está montado en un gran número de archivos, con lo cual es muy difícil seguir el rastro de todos los elementos, una de las facilidades que nos da para controlar de mejor forma esto sería configurarlo de tal forma:

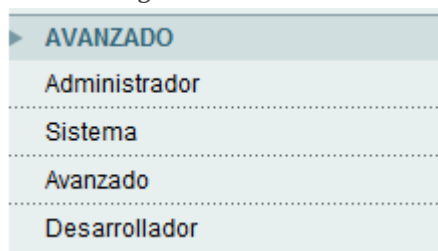
Nos vamos a la parte de Sistema/Configuración:

Figura 44. Sistema/Configuración.



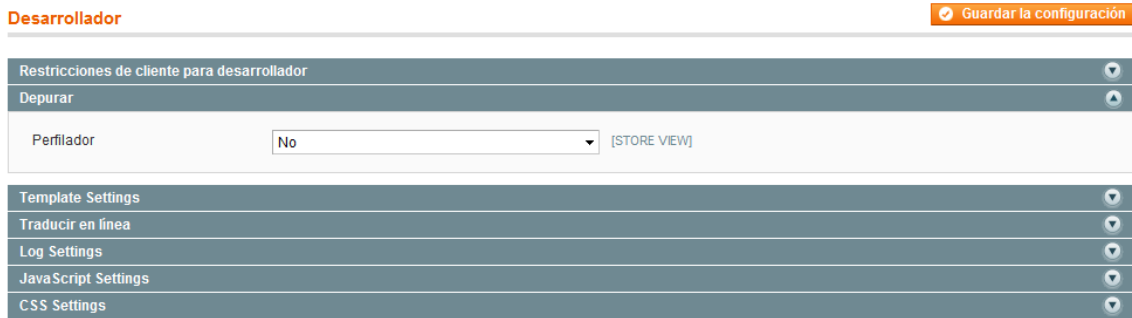
Una vez aquí, abajo del todo a la izquierda, está el modulo Avanzado de Configuración y pulsaremos en la parte "Desarrollador":

Figura 45. Módulo desarrollador de Magento.



Accediendo a la siguiente configuración:

Figura 46. Módulo desarrollador de Magento 2.



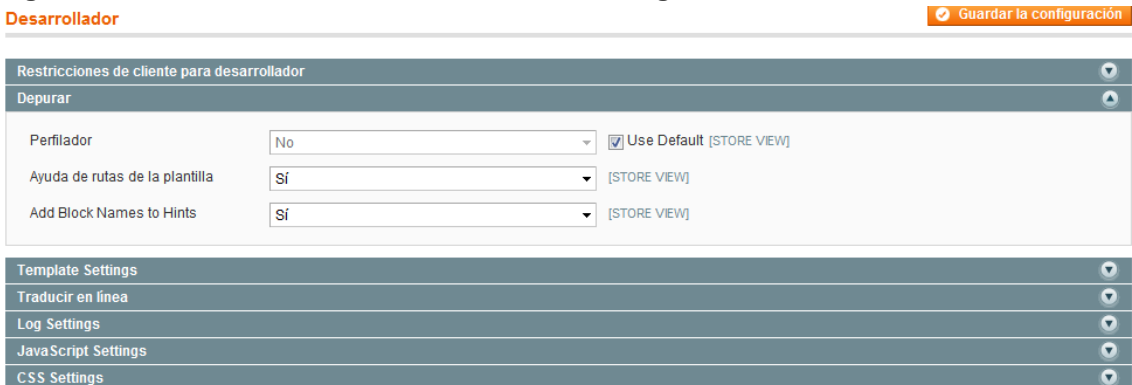
Ya que no es suficiente para lo que queremos activar, cambiaremos el “Alcance de la configuración actual” para ponerlo sobre la tienda y no por defecto:

Figura 47. Cambio de vista para aplicar las propiedades.



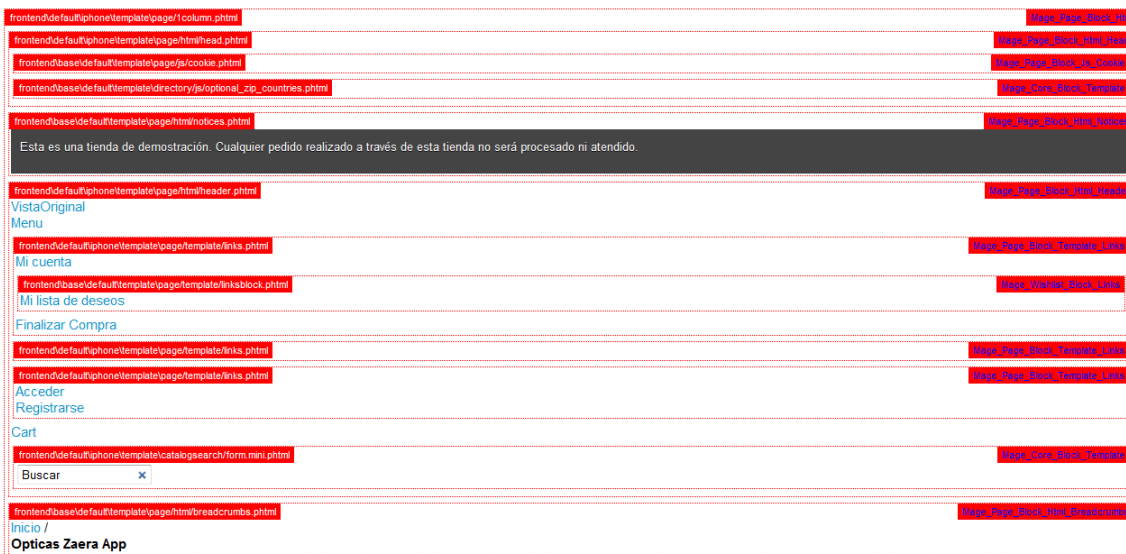
Y por último, se modifica para dejar la configuración de la siguiente forma:

Figura 48. Activación del módulo desarrollador de Magento.



Con esto conseguimos que al cargar cualquiera de las páginas Web dentro de la tienda, se vean de la siguiente forma:

Figura 49. Referencia de los diferentes contenedores de la Web.



En el que se puede identificar en que directorio se encuentra el archivo al que pertenece y en el que está implementado cada elemento y así poder realizar las modificaciones pertinentes de forma más guiada e intuitiva.

La opción de añadir nuevos elementos también es más fácil gracias a esto, dado que se puede saber si tiene que ir dentro de alguno de los contenedores que se marcan como cada archivo (en este caso “phtml”), como un nuevo elemento y la posición donde se va a colocar visualmente.

3.5.4 Funcionalidades

Si se quieren añadir funcionalidades nuevas a la aplicación no es tan fácil como parece, ya que aunque sea más sencillo pudiendo activar el módulo anteriormente mencionado, la mayoría de estos archivos van incluidos en variables globales o bien variables que acceden a la base de datos y con ello recogen la información que se quiere incluir en cada momento.

Al margen de esto, no es imposible implementar o modificar cosas sobre el código. Un ejemplo de nueva funcionalidad puede ser incluir un enlace en los dos diseños de la Web, dando opción a los usuarios de poder cambiar de diseño si así lo desean, principalmente si lo que quieren es cambiar es acceder desde un dispositivo móvil a la versión original de la Web, ya que esta suele ser más completa.

Para hacer esto posible en una tienda como esta, montada en magento:

1. Se ha optado por la opción de que se reconozca por si solo el dispositivo que accede, con la excepción que antes se incorporó a la tienda. Con esto se consigue mostrar diseños diferentes para el tipo de dispositivo, pero perjudica si se quiere dar la opción de cambiar al usuario.
2. Habría que crear una nueva página desde el backend de la tienda de la siguiente forma:

Figura 50. Creación página app desde el backend.

Título	URL Key	Diseño	Vista de tienda	Estado	Fecha de creación	Última modificación	Acción
About Us	about-magento-demo-store	1 column	Todas las vistas de tienda	Habilitado	15/04/2014 20:29:00	11/05/2014 17:47:11	Previsualizar
Opticas Zaera App	app	1 column	Opticas Zaera Main WebSite Opticas Zaera Main Website Store Español	Habilitado	03/09/2014 18:22:35	04/09/2014 21:53:37	Previsualizar
Customer Service	customer-service	3 columns	Todas las vistas de tienda	Habilitado	15/04/2014 20:29:00	11/05/2014 18:05:36	Previsualizar
Enable Cookies	enable-cookies	1 column	Todas las vistas de tienda	Habilitado	15/04/2014 20:29:01	11/05/2014 18:08:27	Previsualizar
Opticas Zaera Home page	home	2 columns with right bar	Opticas Zaera Main WebSite Opticas Zaera Main Website Store Inglés	Habilitado	15/04/2014 20:29:00	17/05/2014 17:55:47	Previsualizar
Opticas Zaera	home	2 columns with right bar	Opticas Zaera Main WebSite Opticas Zaera Main Website	Habilitado	11/05/2014 17:40:34	11/05/2014 17:42:44	Previsualizar

3. Si se suprime la excepción, se puede incluir un enlace en cualquier parte de la app web preferiblemente al final de esta, en el archivo:

/app/design/frontend/default/iphone/template/page/html/footer.phtml

En el que bastaría con incluir un nuevo link/botón/combo...con el que se incluiría el enlace a la dirección de la nueva página creada anteriormente.

4. Se configurará el diseño para la nueva página para que aplique el que se crea para la aplicación.
5. Una vez se ha cambiado a la vista original desde el dispositivo móvil, bastaría con añadir un enlace en la vista original, por ejemplo en la parte superior, accediendo al siguiente archivo:

/public_html/app/design/frontend/base/default/template/page/template/links.phtml

Y añadiendo algo como:

Figura 51. Creación de enlace a la app en el código.

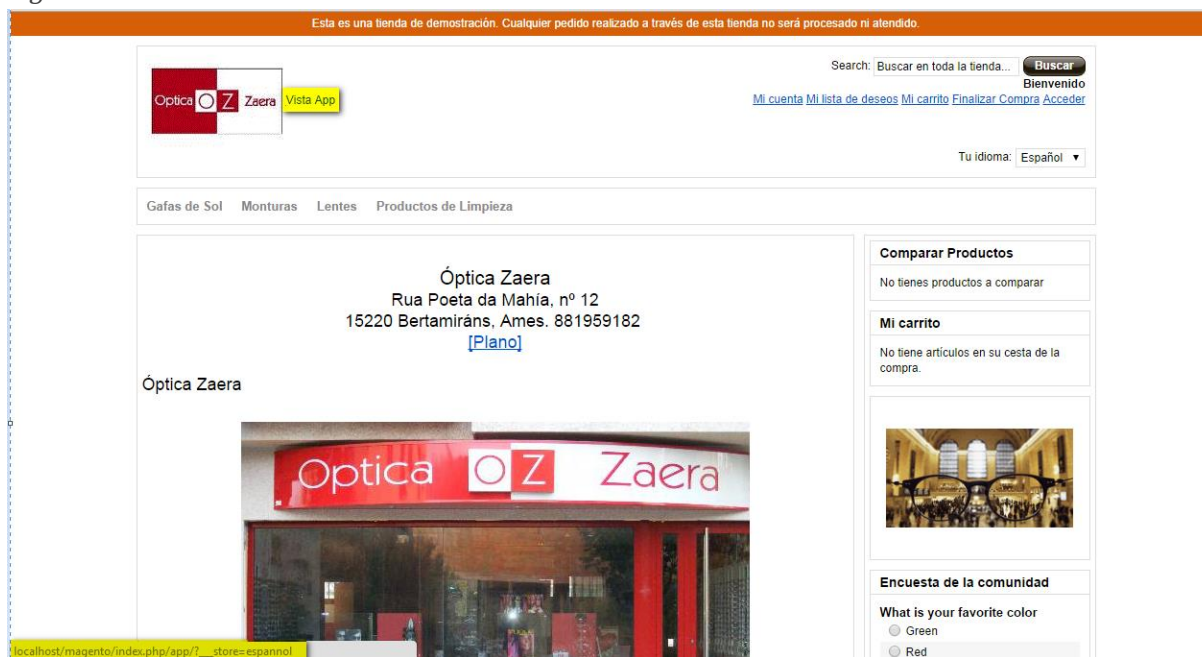
```

28  /**
29   * @see Mage_Page_Block_Template_Links
30   */
31  ?>
32  <?php $links = $this->getLinks(); ?>
33  <?php if(count($links)>0): ?>
34  <ul class="links"<?php if($this->getName(): ?> id="<?php echo $this->getName() ?>"<?php endif;?>
35      <?php foreach($links as $link): ?>
36          <?php if ($link instanceof Mage_Core_Block_Abstract):?>
37              <?php echo $link->toHtml() ?>
38          <?php else: ?>
39              <li?>?php if($link->getIsFirst()||$link->getIsLast()): ?> class="<?php if($link->getIsFirst(): ?>first<?php endif; ?><?php if($link->getIsLast(): ?>last<?php endif; ?>"<?php endif;?>
40              <?php endif;?>
41          <?php endforeach; ?>
42  </ul>
43  <div class="VistaApp">
44      <a href="http://localhost/magento/index.php/app/?__store=espanol">Vista App</a>
45  </div>
46  <?php endif; ?>
47  </pre>

```

Quedando de tal forma:

Figura 52. Muestra del enlace.



De esta forma, cuando el usuario pinche sobre el enlace le llevará de nuevo a la vista de la aplicación.

3.5.5 Formato y diseño

Lo primero que se hace para poder trabajar mejor es suprimir las excepciones anteriormente creadas, y pedir a magento que nos abra por defecto el formato "iphone", con esto conseguiremos que en vez de ver la tienda en formato "default" desde el pc, podremos ver la tienda con el formato App-Web que estamos modificando.

Utilizando una de las utilidades del navegador Chrome, que es "Inspeccionar Elemento", con el que se puede ver el código HTML de cada uno de los elementos. Solamente con pasar el cursor por encima de las líneas de código, este va marcando a que elemento corresponde cada una de esas líneas.

Por otro lado, también se puede seguir el código CSS de todos los elementos, apareciendo su código con todas las propiedades aplicadas.

Dentro de cada una de esas líneas se puede editar, añadiendo o quitando sentencias y ver el efecto que tienen instantáneamente en la misma vista de la aplicación que se muestra en el navegador, incluso se pueden activar o desactivar elementos.

Empezaremos cambiando la parte de arriba para darle nuestro propio formato a la app. Estas especificaciones se encuentran dentro del archivo:

```
\skin\frontend\default\iphone\css\iphone.css
```

En primer lugar se va a crear una sola imagen para la parte de arriba, en la que incluya el logo, en el css actual de la aplicación, lo que incluye es por una parte un fondo y se le añade el logo a su izquierda. En este caso vamos a suprimir el logo actual en el que de momento se dejará comentado por si posteriormente se quisiera añadir el fondo y un

logo superpuesto por separado. Son dos tipos de logo, las líneas a comentar son las siguientes:

Figura 53. Modificación en el código para el logo.

```
/* .header-bg .header-logo { background:url(..images/custom/bg_logo.png) no-repeat 0 0; display:block; height:45px; position:absolute; width:100%;
```

Y para el otro logo:

Figura 54. Modificación en el código para el logo 2.

```
/* .header-bg .header-logo { background-image:url(..images/custom/bg_logo_retina.png); background-size:45px 45px; }*/
```

Ahora, para crear la imagen superior con el logo y el nombre de la tienda se utiliza Adobe PhotoShop CS6, ya que es lo más cómodo para darle un buen formato trabajando por capas. Para ver como se ha creado, se anexa una descripción en el siguiente enlace: [Creación imagen superior.](#)

Bien, una vez ya tenemos la imagen, vamos a modificar código para que cuando la app se cargue, la imagen tenga una altura fija, pero el ancho se adapte dependiendo del dispositivo. Habrá que hacerlo para los dos casos, como ha ocurrido antes con el logo:

Figura 55. Cambio estilo app.

```
.header-bg {  
    background:url(..images/custom/bg_header.png) no-repeat center #fff;  
    height:45px;  
    background-size:100% 45px;  
    -webkit-user-select:none;  
    -webkit-transform:translate3d(0, 0, 1px);  
    -webkit-transform-style:preserve-3d;  
    -webkit-tap-highlight-color:rgba(0,0,0,0);  
    -webkit-touch-callout:none;  
}
```

Y para el otro header:

Figura 56. Cambio estilo app 2.

```
.header-bg { background:url(..images/custom/bg_header_retina.png) no-repeat center; background-size:100% 45px; }
```

3.5.5.1 Estilos

Existen páginas Web, que con solo introducir la URL nos facilitan un estudio de los colores de nuestra página, un ejemplo de estas es la siguiente:

Figura 57. Página análisis de estilos.



Una vez hecho el estudio de colores de la página, podemos ver en detalle cada uno de los componentes analizados:

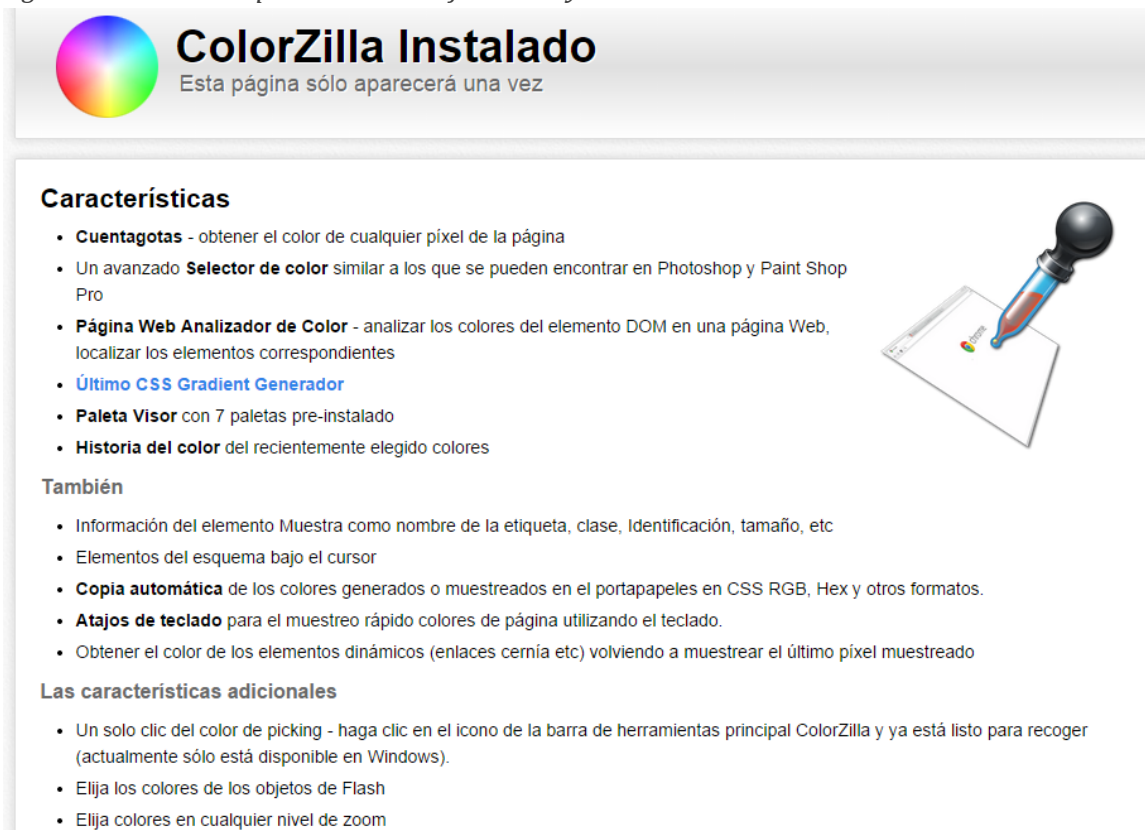
Figura 58. Resultado análisis de estilos.

Node	Foreground	Background	Sample	Contrast Ratio	Brightness difference	Color difference
BODY class='cms-index-index cms-home'	#000000	#FFFFFF	Sample Text	21:1 AAA	255	765
* DIV class='wrapper'	#000000	#FFFFFF	Sample Text	21:1 AAA	255	765
** DIV class='global-site-notice noscri...'	#2F2F2F	#FFF90	Sample Text	12.72:1 AAA	195	513
*** DIV class='notice-inner'	#2F2F2F	#FFF90	Sample Text	12.72:1 AAA	195	513
**** P	#2F2F2F	#FFF90	Sample Text	12.72:1 AAA	195	513
***** STRONG	#2F2F2F	#FFF90	Sample Text	12.72:1 AAA	195	513
***** BR	#2F2F2F	#FFF90	Sample Text	12.72:1 AAA	195	513
** DIV class='global-site-notice demo-n...'	#FFFFFF	#D75F07	Sample Text	3.8:1	134	448
*** DIV class='notice-inner'	#FFFFFF	#D75F07	Sample Text	3.8:1	134	448
**** P	#FFFFFF	#D75F07	Sample Text	3.8:1	134	448

Para poder crear y editar estilos se ha optado por instalar una Extensión para el navegador Google Chrome (también disponible para Mozilla Firefox). Con esta herramienta es muy fácil saber que colores se están utilizando, exportar directamente a

CSS y otros formatos, y muchas cosas más que se detallan al instalarla como se muestra en la siguiente imagen:

Figura 59. Extensión para edición de formatos y estilos.



ColorZilla Instalado
Esta página sólo aparecerá una vez

Características

- **Cuentagotas** - obtener el color de cualquier píxel de la página
- Un avanzado **Selector de color** similar a los que se pueden encontrar en Photoshop y Paint Shop Pro
- **Página Web Analizador de Color** - analizar los colores del elemento DOM en una página Web, localizar los elementos correspondientes
- **Último CSS Gradient Generador**
- **Paleta Visor** con 7 paletas pre-instalado
- **Historia del color** del recientemente elegido colores

También

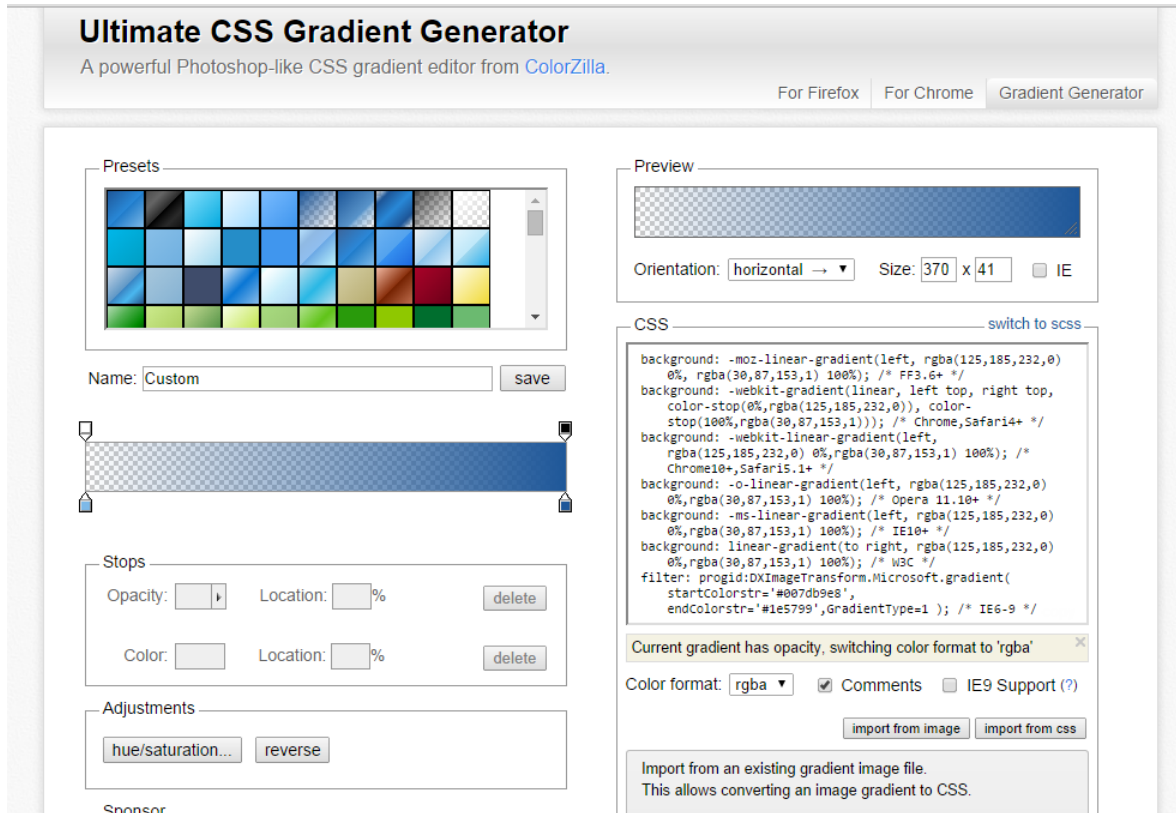
- Información del elemento Muestra como nombre de la etiqueta, clase, Identificación, tamaño, etc
- Elementos del esquema bajo el cursor
- **Copia automática** de los colores generados o muestreados en el portapapeles en CSS RGB, Hex y otros formatos.
- **Atajos de teclado** para el muestreo rápido colores de página utilizando el teclado.
- Obtener el color de los elementos dinámicos (enlaces cernía etc) volviendo a muestrear el último píxel muestreado

Las características adicionales

- Un solo clic del color de picking - haga clic en el icono de la barra de herramientas principal ColorZilla y ya está listo para recoger (actualmente sólo está disponible en Windows).
- Elija los colores de los objetos de Flash
- Elija colores en cualquier nivel de zoom

Por otro lado está disponible un CSS Gradient Generador, para elegir degradados de diferentes tipos y colores y según se configura todo nos va generando el código CSS (también opción en SCSS):

Figura 60. Generador de código CSS.



Para cambiar el color general de la aplicación no se encuentra en el archivo CSS común donde se implementan el resto de estilos de la app, este se encuentra en el mismo directorio:

`\skin\frontend\default\iphone\css\custom.css`

En el que se aplica el color por defecto al MAIN, como se puede observar quedaría como lo siguiente:

Figura 61. Cambio del color general de la aplicación.

```
/* Main color */
body > header,
body > footer ul,
body > address.copyright { background-color: /*#e76212*/ #cc3131; }
```

Con esto se consigue que la aplicación tome ese color general, quedando de la siguiente forma:

Volviendo al archivo principal donde se definen los estilos para el diseño de la App:

`\skin\frontend\default\iphone\css\iphone.css`

Se hace el siguiente cambio en la parte del “nav-container”, para aplicar el siguiente color de tal forma:

Figura 62. Se aplica el color al menú de categorías.

```
#nav-container li a {
background-color: -webkit-gradient(linear, left top, left bottom, color-stop(0%,#b4e391), color-stop(50%,#61c419), color-stop(100%,#b4e391));
background-color: #DDDDDD !important;
display:block;
}
```

Con conseguimos que la aplicación esté bien configurada y atractiva visualmente.

4 Pruebas

Vamos a echar un vistazo a un ejemplo que utiliza este enfoque en el contexto de una aplicación de ejemplo que muestra el estado de la red.

4.1 Pruebas referentes a la aplicación web desarrollada para la tienda

4.1.1 Pruebas unitarias

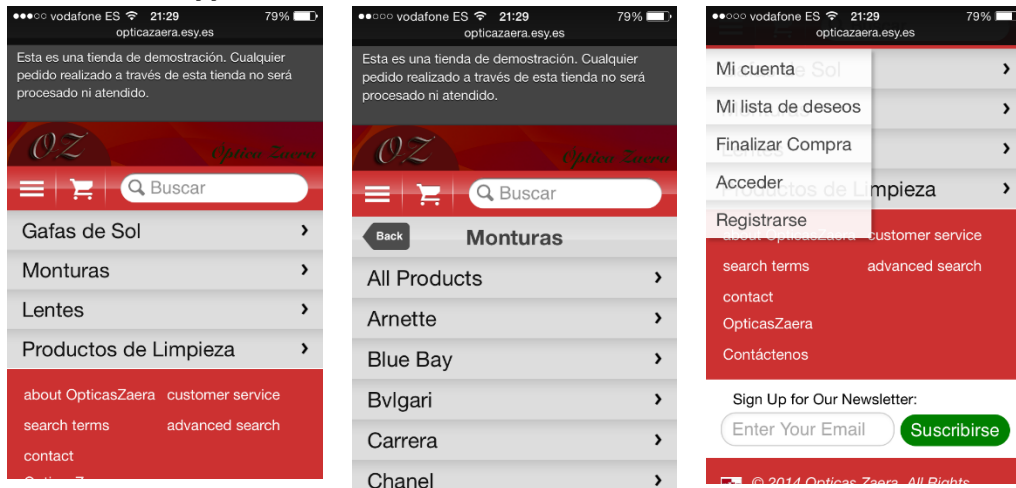
Se prueba la funcionalidad en tres dispositivos diferentes, el detalle y las evidencias de la navegación quedan plasmadas en el “Manual de usuario Aplicación web” adjunto. El dispositivo es un *Smartphone Xiaomi MI3w con Android 4.4.2*.

A continuación se muestra que funciona en dispositivos que trabajan sobre diferentes Sistemas Operativos.

4.1.1.1 Smartphone iPhone 4 con iOS 7

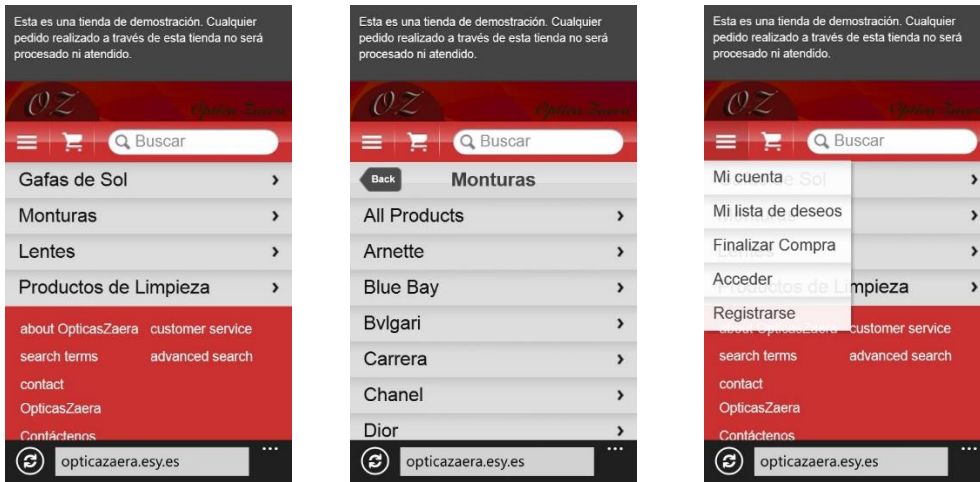
Acceso a la tienda:

Figura 63. Prueba app en iOS 7.



4.1.1.2 Smartphone Nokia Lumia con Windows Phone

Figura 64. Prueba app en Windows Phone 8.0.



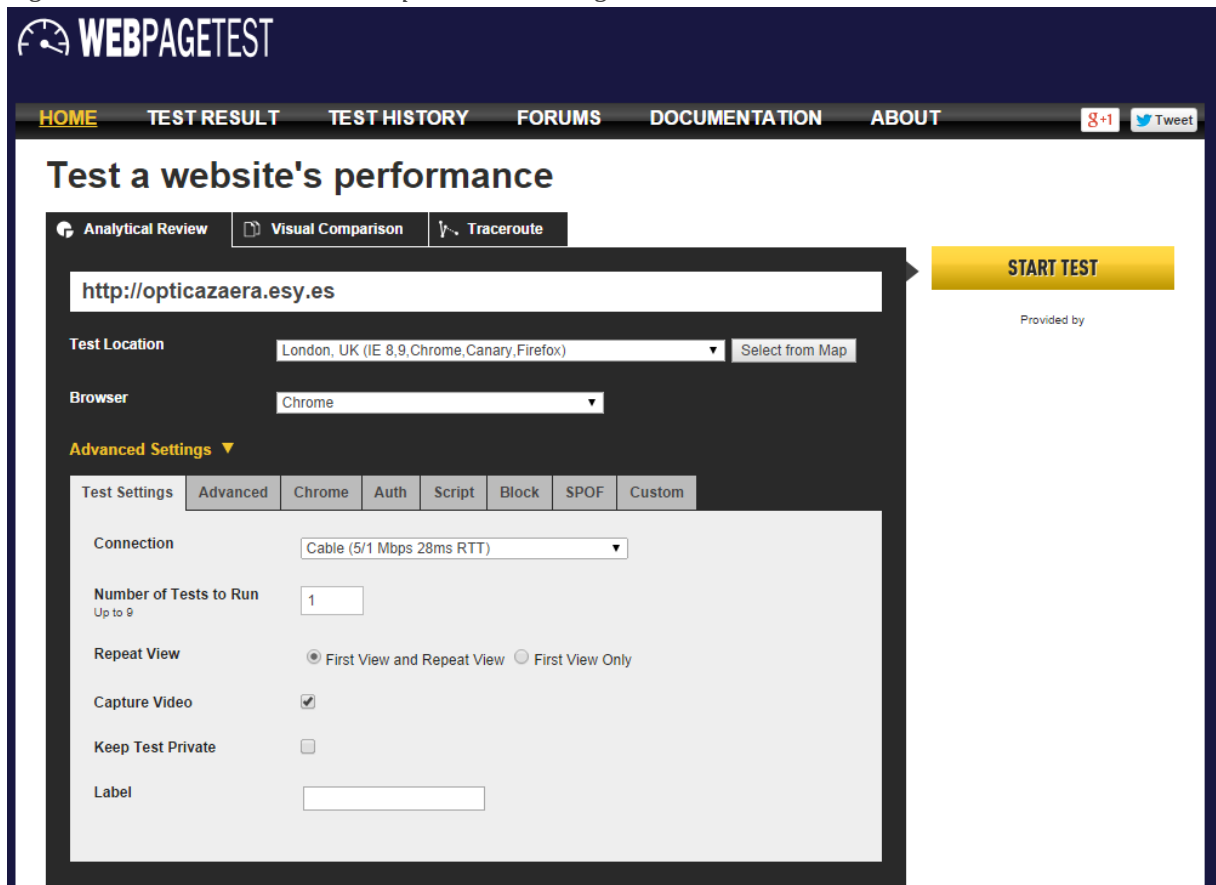
4.1.2 Pruebas de usabilidad

Para este tipo de pruebas vamos a usar una herramienta en línea, que al introducir la URL de la tienda analizará diferentes factores: <http://www.webpagetest.org/>

4.1.2.1 Test tienda "Vista Original"

Lo configuramos de la siguiente forma:

Figura 65. Prueba de usabilidad para la vista original de la tienda.



Después de esperar unos minutos, nos devuelve los resultados del análisis:

Figura 66. Resultados de la prueba de usabilidad para la vista original de la tienda.

Web Page Performance Test for

opticazaera.es

From: London, UK - Chrome - Cable
21/9/2014 20:16:56

F	A	F	A	F	X
First Byte Time	Keep-alive Enabled	Compress Transfer	Compress Images	Cache static content	Effective use of CDN

Summary Details Performance Review Content Breakdown Domains Screen Shot

Re-run the test

[Raw page data - Raw object data](#)
[Export HTTP Archive \(.har\)](#)
[See in ShowSlow](#)
[View Test Log](#)

	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Document Complete			Fully Loaded		
						Time	Requests	Bytes In	Time	Requests	Bytes In
First View	12.221s	0.724s	3.218s	3311	360	12.221s	24	1,785 KB	12.403s	25	1,790 KB
Repeat View	1.927s	0.406s	1.509s	1526	360	1.927s	2	21 KB	1.927s	2	21 KB

	Waterfall	Screen Shot	Video
First View (12.221s)			Filmstrip View Watch Video
Repeat View (1.927s)			Filmstrip View Watch Video
Content Breakdown	<div style="display: flex; justify-content: space-around;"> <div> <p>Requests</p> </div> <div> <p>Bytes</p> </div> </div>		

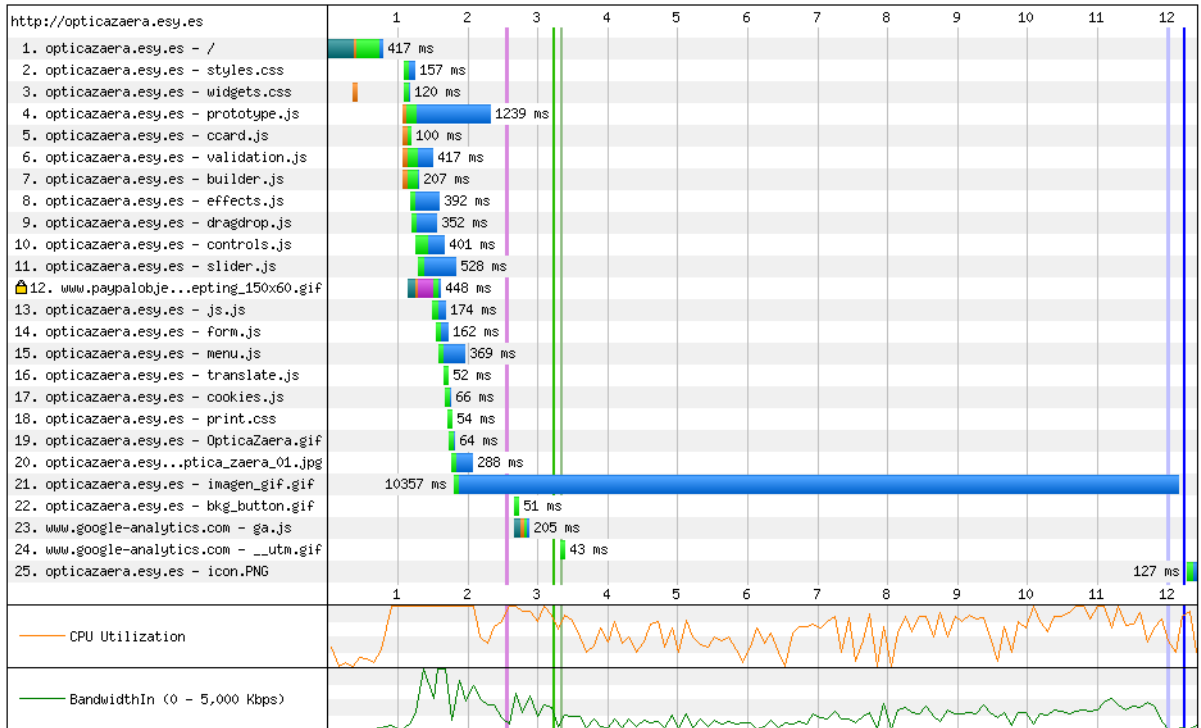
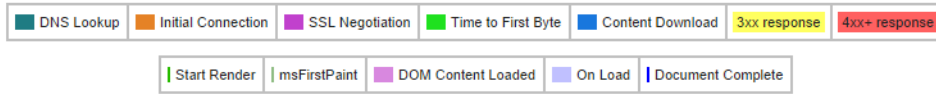
A simple vista se puede observar que si es la primera vez que se carga página han sido 12sg, y el segundo acceso ha sido mucho más rápido con tan solo casi 2sg. Si pinchamos donde indica la flecha se puede ver en detalle los tiempos de para cargar todos los componentes, teniendo en cuenta los archivos a los que accede:

Figura 67. Resultados de la prueba de usabilidad para la vista original de la tienda 2.

Load Time	First Byte	Start Render	Visually Complete	Speed Index	DOM Elements	Result (error code)	Document Complete			Fully Loaded		
							Time	Requests	Bytes In	Time	Requests	Bytes In
12.221s	0.724s	3.218s	9.200s	3311	360	0	12.221s	24	1,785 KB	12.403s	25	1,790 KB

msFirstPaint	domContentLoaded	loadEvent
3.327s	2.544s - 2.579s (0.035s)	11.991s - 12.030s (0.039s)

Waterfall View



[customize waterfall](#) • [View all Images](#) ←

Si pinchamos donde indica la flecha se pueden ver las imágenes que carga la página con sus características:

Figura 68. Lista de las imágenes que contiene la página.

Images are what are currently being served from the given url and may not necessarily match what was loaded at the time of the test.

- 12: https://www.paypalobjects.com/es_ES/i/bnr/bnr_nowAccepting_150x80.gif
0.0 KB image/gif



- 19: <http://opticazaera.esy.es/skin/frontend/base/default/images/OpticaZaera.gif>
4.2 KB image/gif



- 20: http://opticazaera.esy.es/media/favicon/default/optica_zuera_01.jpg
40.6 KB image/jpeg
Baseline (Renders top-down) - Analyze JPEG



- 21: http://opticazaera.esy.es/skin/frontend/default/default/images/media/imagen_gif.gif
1.329.9 KB image/gif



- 22: http://opticazaera.esy.es/skin/frontend/default/blank/images/bkg_button.gif
1.6 KB image/gif



- 24: [http://www.google-analytics.com/__utm.gif?utmwv=5.5.7&utms=1&utm=185289417&utmhn=opticazaera.esy.es&utmcs=UTF-8&utmsr=1280x1024&utmvp=998x834&utmsi=us&utmje=1&utmfi=15.0%20r0&utmdt=Dedicados%20a%20cuidar%20su%20visi%C3%B3n%20Opticas%20Zaera%20Vis%C3%ADtenos&utmhid=1433184598&utmrl=&uti=51548125-1&utmcc=__utma%3D34038991.1319294749.1411323254.1411323254.1411323254.1%3B%2B__utmz%3D34038991.1411323254.1.1.utmcsr%3D\(direct\)%7Cutmcooc%3C](http://www.google-analytics.com/__utm.gif?utmwv=5.5.7&utms=1&utm=185289417&utmhn=opticazaera.esy.es&utmcs=UTF-8&utmsr=1280x1024&utmvp=998x834&utmsi=us&utmje=1&utmfi=15.0%20r0&utmdt=Dedicados%20a%20cuidar%20su%20visi%C3%B3n%20Opticas%20Zaera%20Vis%C3%ADtenos&utmhid=1433184598&utmrl=&uti=51548125-1&utmcc=__utma%3D34038991.1319294749.1411323254.1411323254.1411323254.1%3B%2B__utmz%3D34038991.1411323254.1.1.utmcsr%3D(direct)%7Cutmcooc%3C)
0.0 KB image/gif

- 25: <http://opticazaera.esy.es/media/favicon/default/icon.PNG>
4.9 KB image/png

En una de las imágenes, que aparece resaltado, si clicamos sobre ello, nos lleva a otra página para poder optimizar la imagen y así poder reducir el peso al cargar las páginas:

Figura 69. Optimización de imágenes.

Stats

Image Type: Baseline (Renders top-down)
File Size: 41,548 bytes
Application Data: 4,165 bytes (10.0%)
Optimized Size (Lossless): 35,678 bytes (14.1% smaller)
Quality 85 Size (Lossy): 41,966 bytes (-1.0% smaller)

Image (41,548 bytes)



Optimized Image - Lossless (35,678 bytes)



Quality 85 Image - Lossy (41,966 bytes)



4.1.2.2 Test tienda “Vista Aplicación”

Ahora se hace el mismo procedimiento pero para analizar el funcionamiento de la Aplicación web. Para el análisis se configura de tal forma, cambiando la excepción desde el backend de Magento para que se muestre la app desde el navegador del pc:

Figura 70. Prueba de usabilidad vista de la aplicación.

Test a website's performance

Analytical Review | Visual Comparison | Traceroute

Enter a Website URL

Test Location: London, UK (IE 8,9,Chrome,Canary,Firefox) | Select from Map

Browser: Chrome

Advanced Settings

Test Settings | Advanced | Chrome | Auth | Script | Block | SPOF | Custom

Connection: Mobile 3G - Fast (1.6 Mbps/768 Kbps 150ms)

Number of Tests to Run: 1 (Up to 9)

Repeat View: First View and Repeat View First View Only

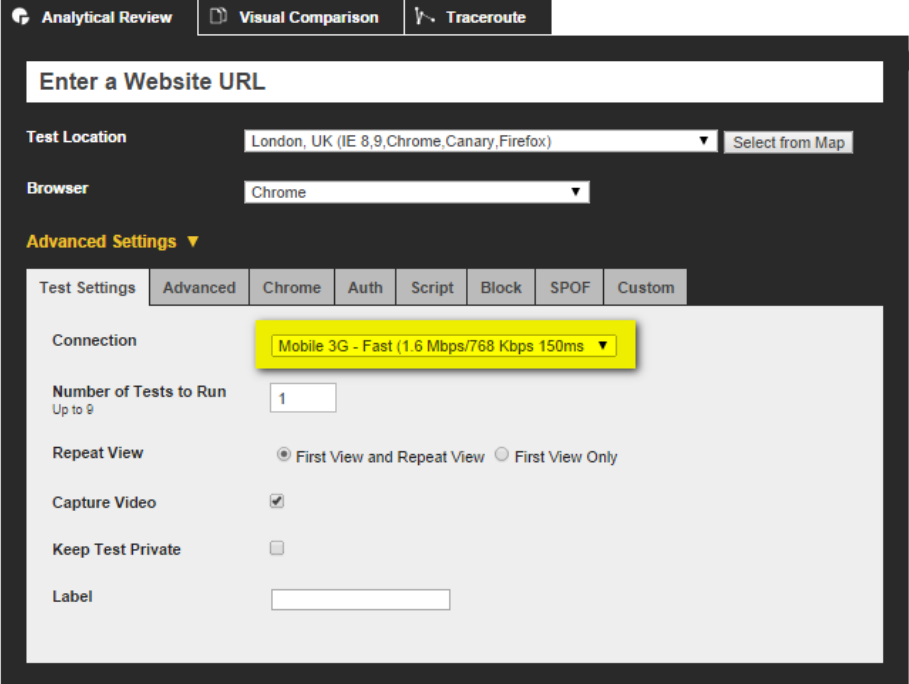
Capture Video:

Keep Test Private:

Label:

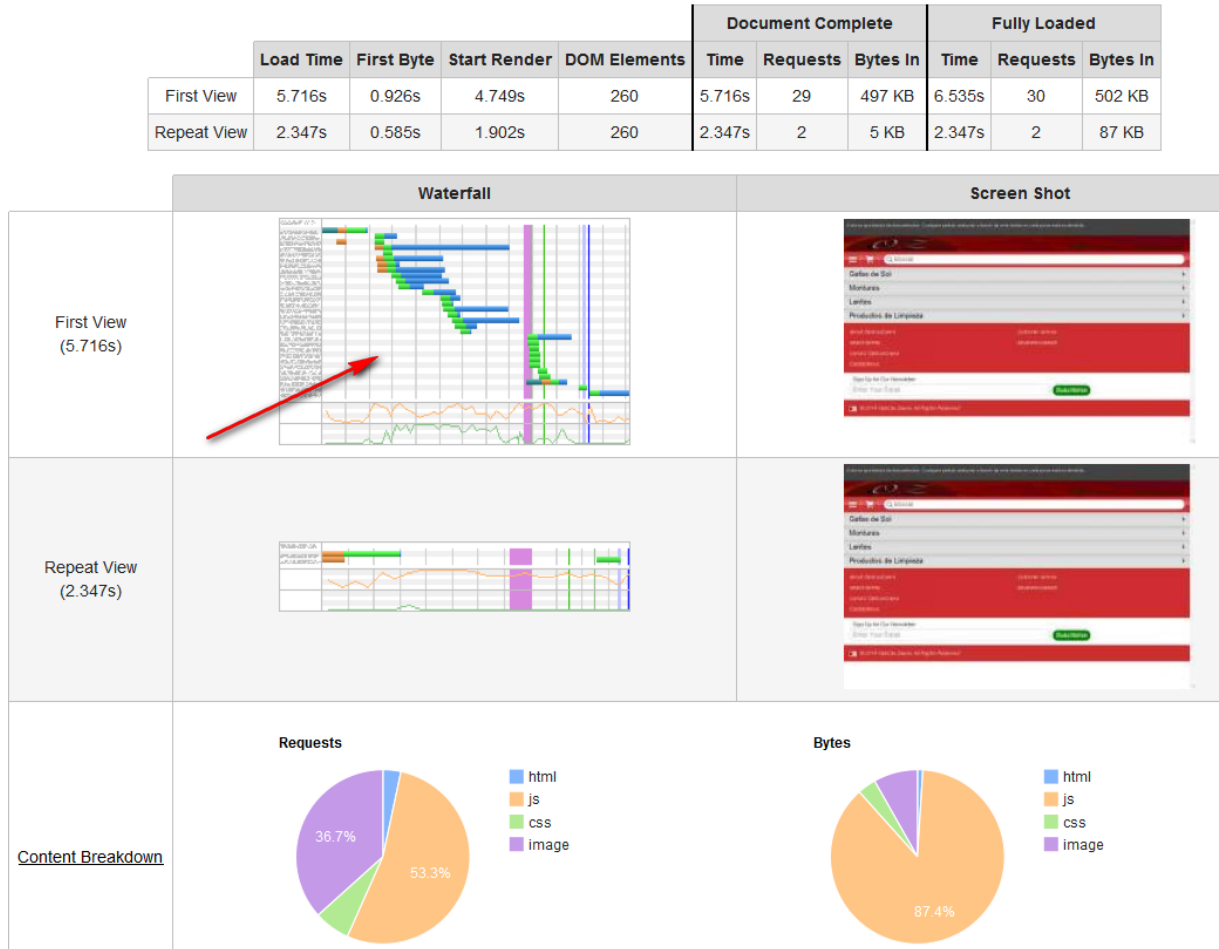
START TEST

Provided by



Se selecciona la conexión para simular de un móvil, después de unos minutos nos da los siguientes resultados:

Figura 71. Resultado de la prueba de usabilidad vista de la aplicación.



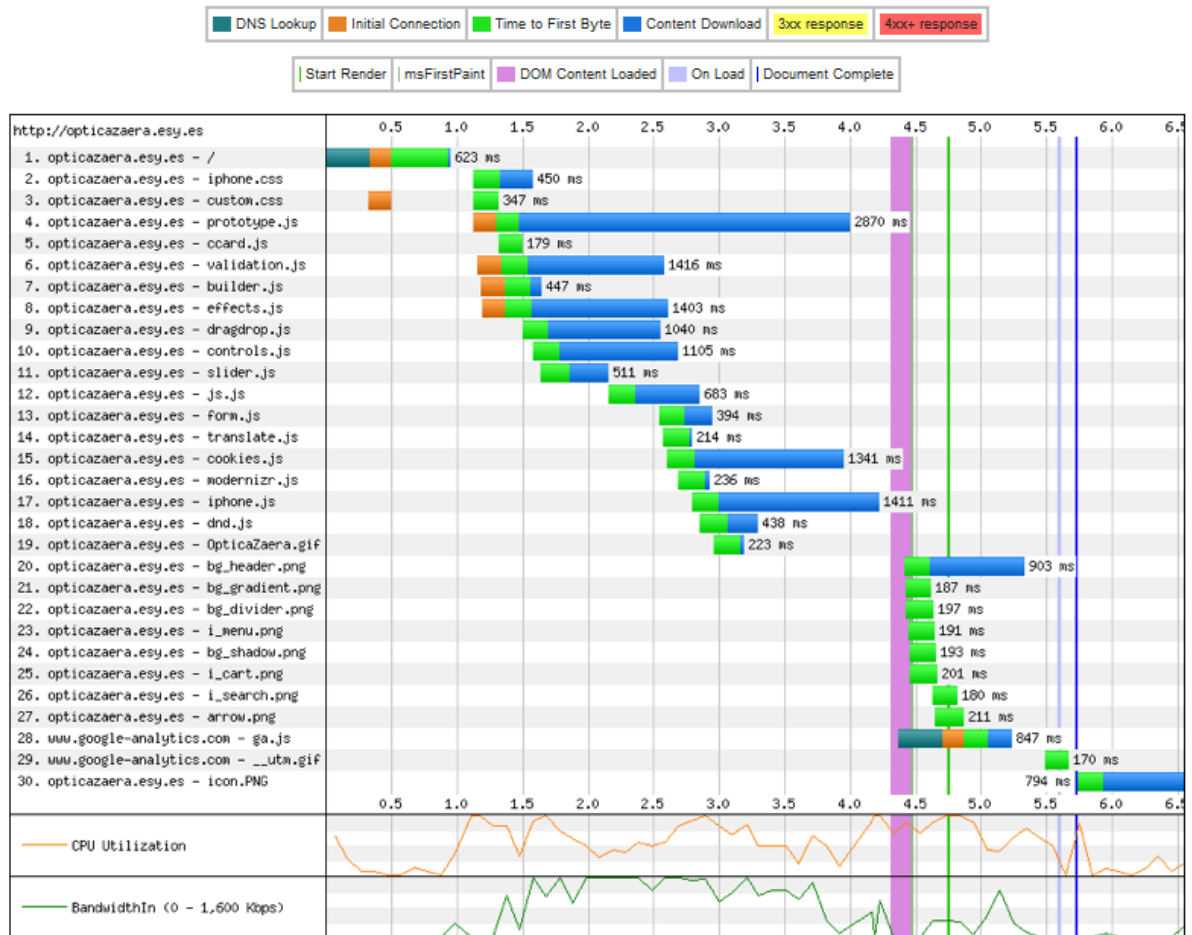
Según se puede observar el tiempo es menos la primera vez que se accede, respecto a la Vista Original, el tamaño total a cargar también es mucho menor. Si entramos al detalle donde indica la flecha:

Figura 72. Resultado de la prueba de usabilidad vista de la aplicación 2.

Load Time	First Byte	Start Render	DOM Elements	Result (error code)	Document Complete			Fully Loaded		
					Time	Requests	Bytes In	Time	Requests	Bytes In
5.716s	0.926s	4.749s	260	0	5.716s	29	497 KB	6.535s	30	502 KB

msFirstPaint	domContentLoaded	loadEvent
4.469s	4.311s - 4.451s (0.140s)	5.588s - 5.607s (0.019s)

Waterfall View



[customize waterfall](#) - [View all Images](#)

Si vamos a ver las imágenes:

Figura 73. Listado de imágenes que carga la vista de la aplicación.

Images are what are currently being served from the given url and may not necessarily match what was loaded at the time of the test.

19: <http://opticazaera.esy.es/skin/frontend/base/default/images/OpticaZaera.gif>
4.2 KB image/gif



20: http://opticazaera.esy.es/skin/frontend/default/iphone/images/custom/bg_header.png
21.7 KB image/png



21: http://opticazaera.esy.es/skin/frontend/default/iphone/images/bg_gradient.png
1.1 KB image/png

22: http://opticazaera.esy.es/skin/frontend/default/iphone/images/bg_divider.png
1.0 KB image/png

23: http://opticazaera.esy.es/skin/frontend/default/iphone/images/i_menu.png
1.0 KB image/png



24: http://opticazaera.esy.es/skin/frontend/default/iphone/images/bg_shadow.png
1.1 KB image/png

25: http://opticazaera.esy.es/skin/frontend/default/iphone/images/i_cart.png
1.6 KB image/png



26: http://opticazaera.esy.es/skin/frontend/default/iphone/images/i_search.png
1.2 KB image/png



27: <http://opticazaera.esy.es/skin/frontend/default/iphone/images/custom/arrow.png>
1.0 KB image/png



29: [http://www.google-analytics.com/__utm.gif?utmwv=5.5.7&utms=1&utmn=2104504246&utmhn=opticazaera.esy.es&utmcs=UTF-8&utmsr=1280x1024&utmvp=998x651us&utmje=1&utmfl=15.0%20r0&utmdl=Opticas%20Zaera&utmhid=1420976501&utmrl=&utmp=%2F&utmht=1411326644959&utmacc=UA-51548125-1&utmcc=__utma%3D34038991.301836095.1411326645.1411326645.1411326645.1%3B%2B__utmz%3D34038991.1411326645.1.1.utmcsr%3D\(direct\)%7Cutmo](http://www.google-analytics.com/__utm.gif?utmwv=5.5.7&utms=1&utmn=2104504246&utmhn=opticazaera.esy.es&utmcs=UTF-8&utmsr=1280x1024&utmvp=998x651us&utmje=1&utmfl=15.0%20r0&utmdl=Opticas%20Zaera&utmhid=1420976501&utmrl=&utmp=%2F&utmht=1411326644959&utmacc=UA-51548125-1&utmcc=__utma%3D34038991.301836095.1411326645.1411326645.1411326645.1%3B%2B__utmz%3D34038991.1411326645.1.1.utmcsr%3D(direct)%7Cutmo)
0.0 KB image/gif

30: <http://opticazaera.esy.es/media/favicon/default/icon.PNG>
4.9 KB image/png



Ahora se puede observar que no nos aconseja optimizar ninguna de ellas. Los resultados parecen buenos en general, aunque se podría mejorar bastante.

5 Conclusiones

Salta a la vista el imparable crecimiento del empleo de dispositivos móviles para poder realizar cada vez más funciones de nuestra vida cotidiana. Si ajustamos más la mira a la parte que engloba el comercio electrónico, nos podemos ubicar en la construcción de negocios online. A lo largo de este Trabajo Fin de Grado se ha explicado el desarrollo desde cero para construir una tienda virtual, llegando al punto de diversificarlo e incluso segmentar el negocio para cubrir una parte importante del mercado, el m-commerce.

Las metas principales a las que se quería llegar han sido del todo posibles, incluso en algunos aspectos sobrepasándolas, llegando a conocer herramientas complementarias que facilitan y optimizan mucho el trabajo, terminando tareas con mejores resultados que si no se hubieran empleado.

Algo de lo que me siento muy orgulloso, los conocimientos adquiridos sobre la herramienta Magento son muy elevados, conociendo en profundidad como funciona, desde el funcionamiento sobre cómo está implementado hasta una gran parte de las infinitas propiedades que tiene de configuración.

Surgieron dificultades, como los conocimientos de lenguajes Web. Antes del desarrollo del Trabajo Fin de Grado, los conocimientos eran algo reducidos, y gracias al haber ahondado en este mundo y habiendo tenido muchos contratiempos para sacar las cosas adelante con buenos resultados, el aprendizaje ha sido muy amplio y favorable.

Otro de los problemas, en un principio, ha sido entender y manejar bien la estructura de directorios, de todos los archivos, ya que está demasiado estructurado. El conocimiento de cada uno de los bloques y el orden que pueden llevar entre ellos. El sistema de archivos que emplea es muy cuantioso y llevar un control de todo es algo complicado.

También es un pequeño obstáculo trabajar bajo un servidor gratuito, ya que no funciona correctamente, es lento y su tamaño, sobre todo el número de inodes (número de archivos), está muy limitado. Esto tiene fácil solución comprando un dominio propio y pagando alguna cuenta Premium en el caso de que se quisiera poner en funcionamiento.

Por descontado, decir que fueron surgiendo algunos errores que impedían por completo el funcionamiento de la tienda y costó muchas horas de esfuerzo encontrar la solución para subsanarlos.

En resumidas cuentas, el haber trabajado en este mundo y con herramientas de este tipo ha sido muy complaciente, principalmente ver que el fruto de todo esto puede ser un negocio totalmente funcional. Además el haber adquirido cierto nivel y conocimiento, en el empleo de estas herramientas es interesante para poder trabajar en este mundo del e-commerce.

Finalmente, he de decir que voy a seguir profundizando en el tema para seguir aprendiendo el uso de las herramientas empleadas y aprender sobre otras herramientas similares, y así tener la posibilidad de crear un negocio propio o trabajar como servicio para empresas o personas que quieran montarlo.

1 Referencias Bibliográficas

SITE GROUND. *Learn more about the different files and folders Magento needs to operate* [en línea]. <<http://www.siteground.com/tutorials/magento/structure.htm>>. [Consulta: 20-04-2014]

CREA TU PROPIA WEB. *Magento: Solution a error 503 Service Temporarily Unavailable The server is temporarily unable to service your request due to maintenance downtime or capacity problems* [en línea].<<http://www.creatupropiaweb.net/cmss/magento-solucion-error-503-service-temporarily-unavailable-server-temporarily-unable-service-request-due-maintenance-downtime-capacity-problems/>>. [Consulta: 20-04-2014]

CAMARA ZARAGOZA. *Como añadir bloques de contenido fácilmente en Magento* [en línea]. <<http://www.camarazaragoza.com/actualidad/como-anadir-bloques-de-contenido-facilmente-en-magento/>>. [Consulta: 30-04-2014]

MAGENTO. *Magento fórum* [en línea]. <<http://www.magentocommerce.com/boards/viewthread/68954/#top>>. [Consulta 28-05-2014]

Noveweb Blog personal de Alfredo Suárez. *Cinco sitios para convertir tu página en una web móvil* [en línea]. <<http://noveweb.wordpress.com/2013/04/22/cinco-sitios-para-convertir-tu-pagina-en-una-web-movil/>>. [Consulta: 12-06-2014]

DESARROLLOWEB.COM. *Qué es Webkit* [en línea]. <<http://www.desarrolloweb.com/articulos/que-es-webkit.html>>. [Consulta: 06-07-2014]

SITE GROUND. *How to enable/disable the demo store notice in Magento* [en línea]. <<http://kb.siteground.com/how-to-enable-disable-the-demo-store-notice-in-magento/>>. [Consulta: 28-07-2014]

CHECK MY COLOURS [en línea]. <<http://www.checkmycolours.com/>>. [Consulta: 04-08-2014]

COLOR ZILLA. *Ultimate CSS Gradient Generator* [en línea]. <<http://www.colorzilla.com/>>. [Consulta: 05-08-2014]

DEVELOPER WORKS. *Android and iPhone browser wars, Part 1: WebKit to the rescue* [en línea]. <<http://www.ibm.com/developerworks/library/os-androidiphone1/>>. [Consulta: 14-08-2014]

DEVELOPER WORKS. *Android and iPhone browser wars, Part 2: Build a browser-based application for iPhone and Android* [en línea]. <<http://www.ibm.com/developerworks/opensource/library/os-androidiphone2/index.html>>. [Consulta: 14-08-2014]

GEEKY THEORY. *¿QUÉ ES Y PARA QUE SIRVE JSON?* [en línea]. <<http://geekytheory.com/json-i-que-es-y-para-que-sirve-json/>>[Consulta: 03-09-2014]

WEB PAGE TEST. *Test a website's performance* [en línea]. <<http://www.webpagetest.org/>>. [Consulta: 15-09-2014]

COMUNIDAD MAGENTO. *Como quitar "Esta es una tienda de demostración..."* [en línea].
<<http://foros.mgnt.es/threads/como-quitar-esta-es-una-tienda-de-demostracion.2229/>>.
[Consulta: 20-09-2014]

NEWEB. *EJECUCIÓN DE MAGENTO INDEXADOR DE SHELL* [EN LÍNEA].
<<http://blog.neweb.co/es/executing-magento-indexer-from-shell/>>. [Consulta: 21-09-2014]

2 Anexos

2.1 Manual desarrollo Magento

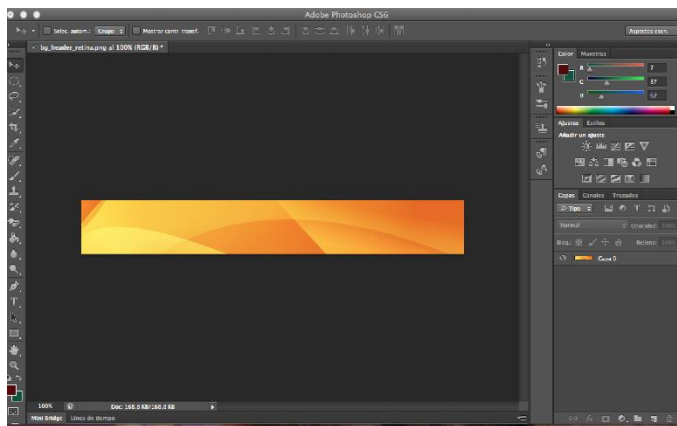
Ir al “Anexo I”

2.2 Manual usuario aplicación web

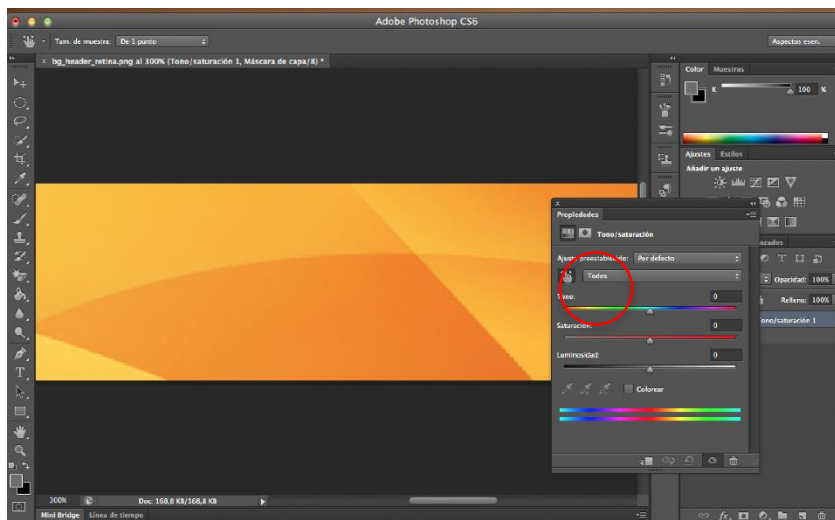
Ir al “Anexo II”

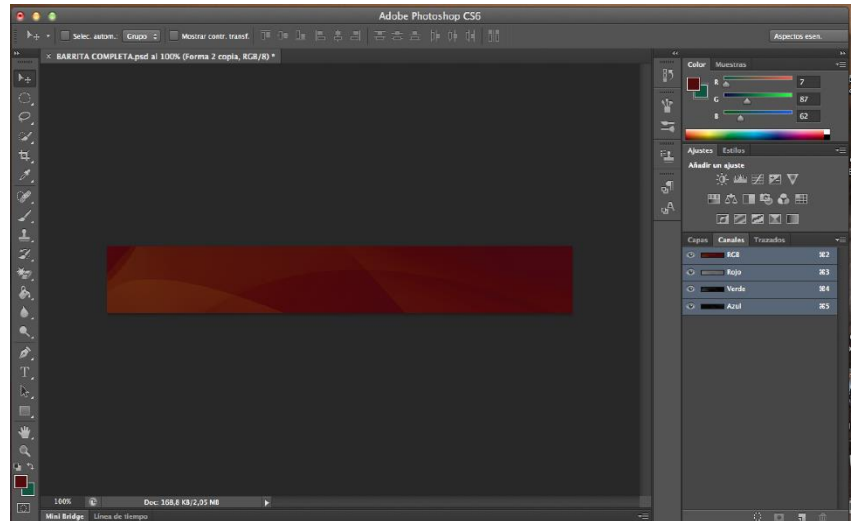
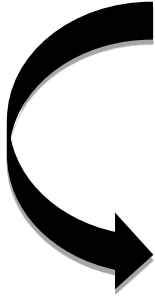
2.3 Creación imagen superior

Para modificar el color inicial de la imagen de cabecera



Hacemos doble click sobre la capa, y nos aparecerá una ventana para seleccionar el tono y la saturación del color del objeto a modificar. Activamos la casilla de selección de imagen (marcada en rojo en pantallazo). Una vez activada, nos podremos mover por el tono, la saturación y la luminosidad, hasta que encontremos el color deseado.

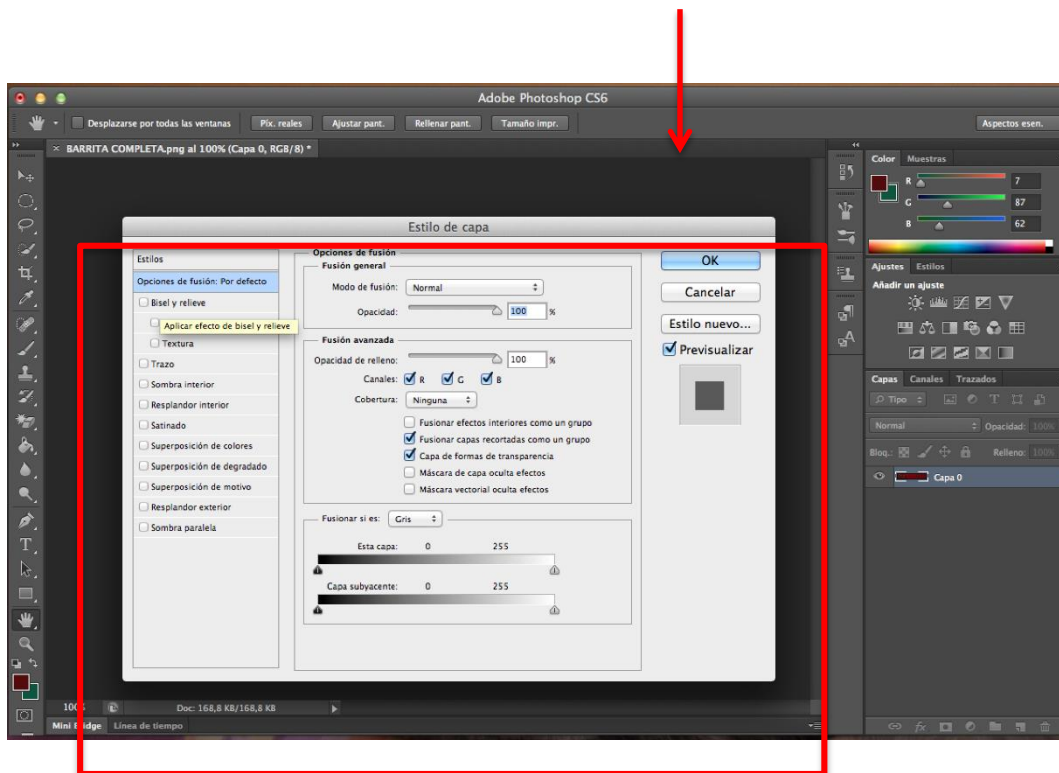




Para crear el nuevo logo de la cabecera

Abrimos un nuevo documento, y le damos las dimensiones que queramos, en este caso de 90x90.

Seleccionamos Herramienta Texto, y le damos el tamaño y tipografía que queramos. Una vez tengamos el texto, hacemos doble click en esta capa y nos aparecerá la siguiente ventana:



Desde aquí podremos cambiar el estilo de las letras, el trazado, los colores de las sombras interiores o exteriores, el degradado, etc.

