

UNIVERSIDAD DE OVIEDO



ESCUELA DE INGENIERÍA INFORMÁTICA

TRABAJO FIN DE MÁSTER

Gestor y generador de tiendas online en el ámbito de las parafarmacias

DIRECTORES: Vicente García Díaz y Begoña Cristina Pelayo García-Bustelo

AUTOR: Pablo Bravo Mediavilla

Agradecimientos

Creo que he llegado al final de mi etapa como estudiante, después de todos estos años, me gustaría agradecer a varias personas todo el apoyo que me han dado.

En primer lugar, agradecer a mis padres todo el esfuerzo que han hecho para pudiera llegar hasta aquí.

Dar las gracias a mi novia, por estar siempre que la he necesitado y ser una pieza fundamental.

A toda mi familia, mis abuelos, tíos, primos y hermanas, que siempre han estado disponibles cuando les he necesitado.

A SimpleLógica, la empresa dónde he realizado las prácticas y que me ha enseñado grandes cosas durante mis meses allí.

A todos los profesores que me han dado clase durante estos años, ya que son otra parte fundamental para llegar hasta aquí y en especial a mis tutores, Cristina Pelayo y Vicente, por toda la ayuda proporcionada y la dedicación.

Por último, agradecer también a mis amigos, al equipo de fútbol sala de la universidad y a los compañeros con los que he trabajado durante estos años por todos los grandes momentos que hemos vivido.

Resumen

Nuestro cliente, dueño de una farmacia, quiere ampliar su área de negocio, para ello, debido al gran auge que está teniendo la venta por internet, quiere tener su parafarmacia online que permita a los usuarios comprar desde cualquier lugar de España. Ya que la mayoría de usuarios utiliza dispositivos móviles cada vez más que los ordenadores, el cliente ha solicitado, como parte importante del proyecto, que la tienda sea usable desde cualquier dispositivo.

Debido al aumento de ventas por internet, nuestro cliente no es el único interesado en montar su propia tienda online, cada día se crean montones de sitios nuevos, la mayoría por gente que apenas tiene conocimientos informáticos y que utiliza gestores de contenidos o desarrolla tiendas muy poco profesionales, debido a la complejidad que ello tiene. Por esta razón, se ha decidido, en primer lugar, desarrollar una parafarmacia online que cumpla con todos los requisitos del cliente, tenga su nombre, logo, productos y el resto de información. Esta parafarmacia se va a desplegar inicialmente en un servidor propio hasta que el cliente solicite su paso a uno más profesional para comenzar a utilizarla.

En segundo lugar, a partir de la tienda online creada, se tratará de externalizar todas las partes comunes a cualquier tienda y crear un generador que permita de manera rápida, escribiendo un simple lenguaje personalizado, tener una nueva aplicación web para la venta de productos completamente operativa. Este generador permitirá, entre otras cosas, modificar el aspecto visual de la tienda, los colores, los menús, el tamaño de los elementos, crear productos o adaptar la información de la página para cumplir con las leyes que afectan al comercio electrónico.

Como complemento para la parafarmacia online, el usuario ha pedido que su gestión sea lo más fácil posible, no quiere tener que pasar imágenes de productos al ordenador y tener que rellenar miles de campos, si no que quiere poder hacerlo desde su dispositivo móvil, que cualquier empleado pueda fácilmente gestionar los productos y los pedidos, así como ver la cantidad de usuarios que se han registrado. Ya que el cliente y todos sus empleados utilizan dispositivos Android, se ha decidido implementar una aplicación para este sistema operativo que permita satisfacer todas sus necesidades.

Finalmente, con la idea de valorar los conocimientos adquiridos durante el máster, se va a añadir un apartado que realice una comparación entre el trabajo fin de grado y el trabajo fin de máster. En este apartado se tratará de ver las diferencias entre ambas interfaces, todas las mejoras que se han incluido, qué tecnologías han cambiado y cualquier otro aspecto que tenga una mejora clara.

Palabras Clave

Parafarmacia, generador, Android, Spring MVC, lenguaje de dominio específico (DSL), Servicios web.

Índice General

CAPÍTULO 1. MEMORIA DEL PROYECTO.....	17
1.1 MOTIVACIÓN	17
1.2 OBJETIVO	18
1.3 ALCANCE.....	19
CAPÍTULO 2. INTRODUCCIÓN.....	21
2.1 ESTUDIO DE LA SITUACIÓN ACTUAL	21
2.2 EVALUACIÓN DE ALTERNATIVAS	22
2.2.1 <i>Parafarmacia Web</i>	22
2.2.2 <i>Cliente Android</i>	24
2.2.3 <i>Generador</i>	25
2.2.4 <i>Bases de datos</i>	26
CAPÍTULO 3. ASPECTOS TEÓRICOS.....	29
3.1 LENGUAJES DE DOMINIO ESPECÍFICO (DSL)	29
3.2 SERVICIO WEB SOAP.....	30
CAPÍTULO 4. GESTIÓN DE PROYECTO.....	31
4.1 PLANIFICACIÓN.....	31
4.1.1 <i>Planificación inicial</i>	31
4.1.2 <i>Planificación inicial detallada</i>	33
4.1.3 <i>Problemas encontrados</i>	37
4.1.4 <i>Planificación final real</i>	37
4.2 PRESUPUESTO	38
4.2.1 <i>Presupuesto resumido</i>	38
4.2.2 <i>Presupuesto detallado</i>	38
4.3 GESTIÓN DE RIESGOS	40
4.3.1 <i>Riesgos identificados</i>	41
CAPÍTULO 5. ANÁLISIS	43
5.1 REQUISITOS DEL SISTEMA	43
5.1.1 <i>Parafarmacia Web</i>	43
5.1.2 <i>Generador</i>	47
5.1.3 <i>Cliente Android</i>	48
5.1.4 <i>Identificación de Actores del Sistema</i>	49
5.1.5 <i>Especificación de Casos de Uso</i>	50
5.2 IDENTIFICACIÓN DE LOS SUBSISTEMAS EN LA FASE DE ANÁLISIS	55
5.2.1 <i>Descripción de los Subsistemas</i>	55
5.2.2 <i>Descripción de los Interfaces entre Subsistemas</i>	58
5.3 ANÁLISIS DE INTERFACES DE USUARIO.....	59
5.3.1 <i>Descripción de la Interfaz</i>	59
5.3.2 <i>Diagrama de Navegabilidad</i>	63
5.4 ESPECIFICACIÓN DEL PLAN DE PRUEBAS.....	65
5.4.1 <i>Pruebas unitarias</i>	65
5.4.2 <i>Pruebas de integración y sistema</i>	67

5.4.3	<i>Pruebas de usabilidad</i>	68
5.4.4	<i>Pruebas de accesibilidad</i>	72
5.4.5	<i>Pruebas de rendimiento</i>	74
CAPÍTULO 6.	DISEÑO DEL SISTEMA	75
6.1	ARQUITECTURA DEL SISTEMA.....	75
6.1.1	<i>Diagramas de Paquetes</i>	79
6.1.2	<i>Diagramas de Despliegue</i>	90
6.2	DISEÑO DE CLASES	90
6.2.1	<i>Diagrama de Clases</i>	90
6.3	DISEÑO DE LA BASE DE DATOS	95
6.4	DISEÑO DE LA INTERFAZ.....	96
6.4.1	<i>Parafarmacia Web</i>	96
6.4.2	<i>Cliente Android</i>	104
6.4.3	<i>Generador</i>	110
CAPÍTULO 7.	IMPLEMENTACIÓN DEL SISTEMA	111
7.1	ESTÁNDARES Y NORMAS SEGUIDOS	111
7.1.1	<i>W3C</i>	111
7.1.2	<i>Android Design</i>	111
7.1.3	<i>Estándares de programación</i>	111
7.1.4	<i>Ley Orgánica de Protección de Datos de Carácter Personal</i>	111
7.1.5	<i>Ley de Servicios de la Sociedad de Información</i>	112
7.2	LINGÜAJES DE PROGRAMACIÓN	113
7.2.1	<i>JAVA</i>	113
7.2.2	<i>HTML 5</i>	114
7.2.3	<i>CSS 3</i>	115
7.2.4	<i>JavaScript</i>	116
7.2.5	<i>PHP</i>	116
7.3	HERRAMIENTAS Y PROGRAMAS USADOS PARA EL DESARROLLO	117
7.3.1	<i>Eclipse</i>	117
7.3.2	<i>Spring Tool Suite</i>	118
7.3.3	<i>Android Studio</i>	119
7.3.4	<i>Illustrator</i>	120
7.3.5	<i>GitHub</i>	121
7.3.6	<i>PHPMyAdmin</i>	121
7.3.7	<i>Draw.io</i>	122
7.3.8	<i>Xampp</i>	122
7.3.9	<i>XText</i>	123
7.4	CREACIÓN DEL SISTEMA.....	124
7.4.1	<i>Problemas Encontrados</i>	124
7.4.2	<i>Descripción Detallada de las Clases</i>	126
CAPÍTULO 8.	DESARROLLO DE LAS PRUEBAS	137
8.1	PRUEBAS UNITARIAS.....	137
8.1.1	<i>Base de datos</i>	137
8.1.2	<i>Generador</i>	138
8.2	PRUEBAS DE INTEGRACIÓN Y DEL SISTEMA.....	140
8.2.1	<i>Generador - servicio Web</i>	140
8.2.2	<i>Cliente Android – Servicio Web</i>	142
8.3	PRUEBAS DE USABILIDAD Y ACCESIBILIDAD	143

8.3.1	Pruebas de Usabilidad	143
8.3.2	Pruebas de Accesibilidad	162
8.4	PRUEBAS DE RENDIMIENTO	184
8.4.1	PageSpeed	184
8.4.2	PingDom	185
8.4.3	Formulario	186
CAPÍTULO 9.	MANUALES DEL SISTEMA	189
9.1	MANUAL DE INSTALACIÓN Y EJECUCIÓN	189
9.1.1	Parafarmacia Web	189
9.1.2	Cliente Android	191
9.1.3	Generador.....	192
9.2	MANUAL DE USUARIO	194
9.2.1	Parafarmacia Web	194
9.2.2	Cliente Android.....	203
9.3	MANUAL DEL PROGRAMADOR.....	213
9.3.1	Parafarmacia Web	213
9.3.2	Cliente Android.....	214
9.3.3	Generador.....	215
CAPÍTULO 10.	TFM VS TFG.....	217
10.1	PARAFARMACIA WEB	217
10.2	SERVICIOS WEB.....	229
10.3	CLIENTE ANDROID.....	230
10.4	GENERADOR	233
CAPÍTULO 11.	CONCLUSIONES Y AMPLIACIONES.....	234
11.1	CONCLUSIONES	234
11.2	AMPLIACIONES.....	235
11.2.1	Métodos de pago	235
11.2.2	Ofertas para productos	235
11.2.3	Recuperar contraseñas	235
11.2.4	SEO.....	235
CAPÍTULO 12.	REFERENCIAS BIBLIOGRÁFICAS	237
CAPÍTULO 13.	APÉNDICES.....	239
13.1	GLOSARIO Y DICCIONARIO DE DATOS.....	239
13.2	CONTENIDO ENTREGADO DIGITALMENTE	240
13.2.1	Contenidos	240
13.3	ÍNDICE ALFABÉTICO.....	242

Índice de Figuras

Ilustración 1 Ejemplo del uso de beans en Spring	23
Ilustración 2 Tareas del análisis	33
Ilustración 3 Diagrama de Gantt del análisis	33
Ilustración 4 Tareas de documentación	33
Ilustración 5 Diagrama de Gantt de la documentación	34
Ilustración 6 Tareas de desarrollo de la parafarmacia	34
Ilustración 7 Diagrama de Gantt del desarrollo de la parafarmacia	34
Ilustración 8 Tareas de desarrollo del servicio web	34
Ilustración 9 Diagrama de Gantt del desarrollo del servicio web	35
Ilustración 10 Tareas de desarrollo del cliente Android	35
Ilustración 11 Diagrama de Gantt del desarrollo del cliente Android	35
Ilustración 12 Tareas de la preparación del servidor	36
Ilustración 13 Diagrama de Gantt de la preparación del servidor	36
Ilustración 14 Tareas de la fase de cierre del proyecto	36
Ilustración 15 Diagrama de Gantt de la fase de cierre del proyecto.....	36
Ilustración 16 Tareas de documentación reales.....	37
Ilustración 17 Boceto interfaz parafarmacia Web	60
Ilustración 18 Boceto interfaz Cliente Android	61
Ilustración 19 Interfaz generador	61
Ilustración 20 Ayuda para escribir el lenguaje del generador	62
Ilustración 21 Explorador de archivos del generador.....	62
Ilustración 22 Diagrama de navegabilidad parafarmacia web	63
Ilustración 23 Diagrama de navegabilidad cliente Android.....	64
Ilustración 24 Arquitectura genera.....	78
Ilustración 25 Diagrama de paquetes	81
Ilustración 26 Diagrama de paquetes com	82
Ilustración 27 Diagrama de paquetes impl	83
Ilustración 28 Diagrama de paquetes Cliente Android.....	85
Ilustración 29 Diagrama de paquetes Generador	88
Ilustración 30 Diagrama de despliegue.....	90
Ilustración 31 Diagrama de clases parafarmacia web	93
Ilustración 32 Diagrama de clases Cliente Android	94
Ilustración 33 Diagrama entidad relación de la base de datos.....	95
Ilustración 34 Pantalla parafarmacia web inicio	96
Ilustración 35 Pantalla parafarmacia web productos	97
Ilustración 36 Pantalla parafarmacia web carrito	97
Ilustración 37 Pantalla parafarmacia web iniciar sesión	98
Ilustración 38 Pantalla parafarmacia web mis pedidos.....	98
Ilustración 39 Pantalla parafarmacia web actualizar datos	99
Ilustración 40 Pantalla parafarmacia web envío y pago	99
Ilustración 41 Pantalla parafarmacia web inicio responsive	100
Ilustración 42 Pantalla parafarmacia web menú responsive	101
Ilustración 43 Pantalla parafarmacia web productos responsive.....	102
Ilustración 44 Pantalla parafarmacia web carrito responsive	103
Ilustración 45 Pantalla cliente Android splash.....	104

Ilustración 46 Pantalla cliente Android principal.....	105
Ilustración 47 Pantalla cliente Android pedidos.....	106
Ilustración 48 Pantalla cliente Android productos	107
Ilustración 49 Pantalla cliente Android usuarios	108
Ilustración 50 Pantalla cliente Android actualizar producto.....	109
Ilustración 51 Pantalla lenguaje Generador	110
Ilustración 52 Ejemplo de código HTML utilizado en la parafarmacia web junto a JSP	114
Ilustración 53 Ejemplo de código CSS utilizado en la parafarmacia web	115
Ilustración 54 Ejemplo ventana de Eclipse con el lenguaje de generación	117
Ilustración 55 Pantalla inicial Spring Tool Suite con el proyecto parafarmacia	118
Ilustración 56 Pantalla de ejemplo de Android Studio	119
Ilustración 57 Pantalla de ejemplo de Illustrator	120
Ilustración 58 Página principal GitHub con el proyecto.....	121
Ilustración 59 Herramienta Draw con el diagrama de paquetes del cliente Android	122
Ilustración 60 Ejemplo de DSL del generador con Xtext	123
Ilustración 61 Salida esperada base de datos MySQL.....	141
Ilustración 62 Resultado prueba de optimización para móviles	154
Ilustración 63 Prueba de optimización Quirktools	154
Ilustración 64 Parafarmacia web sin imágenes	164
Ilustración 65 Parafarmacia web con el tamaño aumentado	165
Ilustración 66 Parafarmacia web con el tamaño de letra aumentado	166
Ilustración 67 Parafarmacia web versión iPad	167
Ilustración 68 Parafarmacia web versión iPhone	168
Ilustración 69 Parafarmacia web sin hoja de estilos.....	170
Ilustración 70 Parafarmacia web en Opera	171
Ilustración 71 Parafarmacia web en Firefox	171
Ilustración 72 Parafarmacia web en Internet Explorer	172
Ilustración 73 Parafarmacia web en Google Chrome	172
Ilustración 74 Parafarmacia web descripción examinada con WAVE	173
Ilustración 75 Parafarmacia web productos examinada con WAVE	173
Ilustración 76 Parafarmacia web registro examinada con WAVE	174
Ilustración 77 Parafarmacia web examinada con TAW	174
Ilustración 78 Parafarmacia web examinada con HERA	175
Ilustración 79 Parafarmacia web inicio validación HTML	176
Ilustración 80 Parafarmacia web productos validación HTML.....	176
Ilustración 81 Can I Use Variables CSS	177
Ilustración 82 Validador CSS Jigsaw	177
Ilustración 83 Validador CSS CSS-Validator	178
Ilustración 84 Test de rendimiento PageSpeed	184
Ilustración 85 Test de rendimiento PingDom	185
Ilustración 86 Entorno de desarrollo Spring Tool Suite	189
Ilustración 87 Entorno de desarrollo Spring Tool Suite Run As...	190
Ilustración 88 XAMPP con MySQL inicializado	190
Ilustración 89 Ruta de almacenamiento de imágenes.....	191
Ilustración 90 Archivo .apk para instalar nuestro cliente Android.....	191
Ilustración 91 Eclipse Marketplace con Xtext.....	192
Ilustración 92 Eclipse Xtext	193
Ilustración 93 Eclipse generador Xtext	193
Ilustración 94 Pantalla principal parafarmacia web	194
Ilustración 95 Enlace de registro parafarmacia web.....	195

Ilustración 96 Pantalla de registro parafarmacia web	195
Ilustración 97 Formulario de registro parafarmacia web	196
Ilustración 98 Enlace de acceso parafarmacia web	197
Ilustración 99 Acceso parafarmacia web	197
Ilustración 100 Cerrar sesión parafarmacia web	198
Ilustración 101 Modificar cuenta parafarmacia web	199
Ilustración 102 Formulario para modificar cuenta parafarmacia web	199
Ilustración 103 Productos parafarmacia web	200
Ilustración 104 Número productos carrito parafarmacia web	200
Ilustración 105 Carrito parafarmacia web	201
Ilustración 106 Resumen compra parafarmacia web	201
Ilustración 107 Compra finalizada parafarmacia web	202
Ilustración 108 Mis pedidos parafarmacia web	202
Ilustración 109 Pantalla de inicio y principal cliente Android	203
Ilustración 110 Pantalla principal con opción pedidos cliente Android	204
Ilustración 111 Pantalla pedidos cliente Android	204
Ilustración 112 Pantalla pedidos con estado cliente Android	205
Ilustración 113 Pantalla principal con opción usuarios cliente Android	206
Ilustración 114 Pantalla usuarios cliente Android	206
Ilustración 115 Pantalla principal con opción productos cliente Android	207
Ilustración 116 Pantalla productos cliente Android	207
Ilustración 117 Pantalla productos con nuevo producto cliente Android	208
Ilustración 118 Pantalla nuevo producto cliente Android	209
Ilustración 119 Pantalla productos con opción actualizar producto cliente Android	210
Ilustración 120 Pantalla actualizar producto cliente Android	211
Ilustración 121 Pantalla principal Parafarmacia TFG	218
Ilustración 122 Pantalla principal Parafarmacia TFM	218
Ilustración 123 Logos del TFG y TFM	219
Ilustración 124 Carrito del TFG y TFM	219
Ilustración 125 Menús del TFG y TFM	220
Ilustración 126 Uso de migas en el TFM	221
Ilustración 127 Contraste de colores TFG y TFM	222
Ilustración 128 Títulos TFG y TFM	223
Ilustración 129 Formularios TFG y TFM	224
Ilustración 130 Atajos TFM	225
Ilustración 131 Buscador TFG y TFM	226
Ilustración 132 Página principal con cartel de cookies parafarmacia web	227
Ilustración 133 Política de privacidad y cookies parafarmacia web	227
Ilustración 134 Inicio de sesión parafarmacia web con cookies	228
Ilustración 135 Pantalla principal cliente Android TFG y TFM	230
Ilustración 136 Pantalla pedidos cliente Android TFG y TFM	231
Ilustración 137 Pantalla productos cliente Android TFG y TFM	232

Índice de tablas

Tabla 1 Tareas planificadas	32
Tabla 2 Resumen del presupuesto	38
Tabla 3 Coste roles por horas de trabajo	38
Tabla 4 Coste material por horas de trabajo	39
Tabla 5 Coste total por fases.....	39
Tabla 6 Resumen final de costes	39
Tabla 7 Riesgos del proyecto	42
Tabla 8 Requisitos cuenta de usuario parafarmacia web	43
Tabla 9 Requisitos compra de productos parafarmacia web.....	44
Tabla 10 Requisitos no funcionales parafarmacia web	46
Tabla 11 Requisitos funcionales generador.....	47
Tabla 12 Requisitos no funcionales generador.....	47
Tabla 13 Requisitos funcionales cliente Android.....	48
Tabla 14 Requisitos no funcionales cliente Android.....	49
Tabla 15 Pruebas unitarias base de datos	66
Tabla 16 Pruebas unitarias generador	66
Tabla 17 Pruebas de integración y sistema generador - servicios web	67
Tabla 18 Pruebas de integración y cliente Android - servicios web	68
Tabla 19 Cuestionario pruebas usabilidad con usuarios.....	70
Tabla 20 Cuestionario pruebas de accesibilidad.....	73
Tabla 21 Cuestionario pruebas de rendimiento	74
Tabla 22 Clase producto parafarmacia web	126
Tabla 23 Clase pedido parafarmacia web.....	126
Tabla 24 Clase carrito parafarmacia web	127
Tabla 25 Clase usuario parafarmacia web.....	127
Tabla 26 Clase pedido DAO parafarmacia web	128
Tabla 27 Clase producto DAO parafarmacia web	128
Tabla 28 Clase usuario DAO parafarmacia web.....	128
Tabla 29 Clase welcome controller parafarmacia web	129
Tabla 30 Clase acceso controller parafarmacia web.....	129
Tabla 31 Clase actualizar carrito controller parafarmacia web.....	129
Tabla 32 Clase buscador controller parafarmacia web.....	130
Tabla 33 Clase carrito controller parafarmacia web.....	130
Tabla 34 Clase cuenta controller parafarmacia web	130
Tabla 35 Clase detalles controller parafarmacia web.....	131
Tabla 36 Clase finalizar pedido controller parafarmacia web.....	131
Tabla 37 Clase pedidos controller parafarmacia web.....	131
Tabla 38 Clase productos controller parafarmacia web	132
Tabla 39 Clase splash activity cliente Android.....	133
Tabla 40 Clase productos activity cliente Android.....	134
Tabla 41 Clase pedidos activity cliente Android	134
Tabla 42 Clase usuarios activity cliente Android	134
Tabla 43 Clase my DB helper cliente Android.....	135
Tabla 44 Clase pedidos data source cliente Android	135
Tabla 45 Clase productos data source cliente Android.....	136
Tabla 46 Clase usuarios data source cliente Android	136

Tabla 47 Resultado pruebas unitarias base de datos.....	138
Tabla 48 Resultado pruebas de integración cliente Android - servicio web	143
Tabla 49 Resultado cuestionario accesibilidad con usuarios.....	146
Tabla 50 Guía accesibilidad Yusef Hassan Montero.....	153
Tabla 51 Resultado cuestionario accesibilidad	163
Tabla 52 Puntos de verificación prioridad 1 WCAG	180
Tabla 53 Puntos de verificación prioridad 2 WCAG	181
Tabla 54 Puntos de verificación prioridad 3 WCAG	182
Tabla 55 Resultado pruebas de rendimiento.....	186
Tabla 56 Contenido entregado	241

Capítulo 1. Memoria del Proyecto

1.1 Motivación

El proyecto surge a partir de la idea de mi padre para extender su área de negocio, es farmacéutico y tiene una farmacia física. La idea es crear una tienda online que le permita vender medicamentos a través de internet. Tras varias reuniones con el colegio farmacéutico, se decide que vender medicamentos por internet no es rentable, ya que se limita la cantidad y se obliga al farmacéutico a revisar cada pedido con el cliente para comprobar que es realmente lo que necesita. Por esta razón, se descarta esta idea y surge una nueva, vender únicamente productos de parafarmacia, éstos no necesitan receta ni tienen ninguna restricción sobre la cantidad.

Después de tener la idea clara, se comienza con el desarrollo de la primera versión para el trabajo fin de grado, se decide añadir una herramienta de gestión que se pueda instalar en los dispositivos móviles, ya que es lo más cómodo y utilizado por los trabajadores de la farmacia. Además, también se añade un generador de tiendas online que permita facilitar la tarea en el futuro de crear otras parafarmacias para diferentes usuarios que las necesiten.

Por una serie de imprevistos, después de desarrollar el trabajo fin de grado, se abandona la idea de poner en marcha la parafarmacia online.

Hace unos meses vuelve de nuevo a surgir este plan, teniendo como extra todos los conocimientos aprendidos durante el máster. Al evaluar el trabajo desarrollado, me doy cuenta de que el gestor de contenidos utilizado para crear la parafarmacia en el trabajo fin de grado no es tan seguro ni se puede adaptar o personalizar a todas las necesidades, así que decido comenzar un nuevo desarrollo desde cero partiendo de un lenguaje de programación nuevo que permita crear todas las partes sin ayuda de gestores de contenido para tener el control pleno de todo lo que se va a desarrollar.

1.2 Objetivo

El objetivo principal de este proyecto es construir una aplicación web accesible a todos los usuarios de internet, permitiéndoles comprar productos de parafarmacia, así como realizar una gestión completa de la misma por parte de los administradores. Con esta aplicación, se pretende aumentar el número de clientes y con ello, las ventas. Además, para finalizar, se realizará una comparación con el proyecto previamente realizado como trabajo fin de grado, para ver las mejoras que se han incluido y todos los cambios planteados.

Para conseguir este objetivo general, hay que cumplir una serie de objetivos más pequeños dentro del proyecto:

- Construir un generador de aplicaciones Web que permita crear de manera personalizada nuevas tiendas online.
- Construir una aplicación web para la venta de productos que permita a los usuarios:
 - Visualizar una lista de todos los productos.
 - Realizar la compra de uno o varios productos.
- Construir una aplicación para la plataforma Android que permita a los administradores gestionar la aplicación web y obtener información necesaria para la venta de productos:
 - Ver los usuarios de la aplicación.
 - Ver la lista de pedidos solicitados por los usuarios.
 - Gestionar los pedidos.
 - Ver la lista de productos.
 - Gestionar los productos.
- Preparar un servidor web que permita alojar la aplicación web junto a los servicios necesarios para la conexión con el cliente Android.
- Realizar un documento que recoja los cambios y las mejoras que se han realizado en comparación con el proyecto fin de grado.

1.3 Alcance

Se va a desarrollar en primer lugar el generador, que permitirá crear tiendas online personalizadas, modificando la información o la interfaz de la tienda en función de las necesidades del cliente. Este generador se creará a partir de la parafarmacia web y solo permitirá crear otras páginas similares dedicadas a la venta online.

La *parafarmacia web* que se va a desarrollar permitirá a los usuarios registrarse en ella, ver la lista de productos disponibles, añadirlos a un carrito y realizar la compra de productos mediante la opción de pago “*contra reembolso*” o “*pasarela de pago*”, también, recibirán una notificación con el resumen de la compra realizada. Por otro lado, solo se añadirán una serie de productos de prueba en cada una de las categorías, dejando al titular de la web la opción de crear sus propios productos. Para los administradores de la tienda, se permitirá la opción de gestionar los productos, ver los pedidos y usuarios.

Respecto al *cliente Android*, para facilitar aún más el trabajo a los administradores de la tienda, se permitirá ver una lista de todos los usuarios registrados, pero no podrán ser modificados de ninguna forma, los productos podrán ser gestionados, añadir nuevos, eliminar, modificar... Se permitirá ver una lista con los pedidos de los diferentes usuarios y modificar el estado en el que se encuentra el pedido. Este cambio de estado supondrá un aviso al comprador del pedido.

El servidor tendrá una disponibilidad 24/7, salvo contratiempos que se escapen a nuestro alcance y permitirá alojar la aplicación web, así como los servicios web para la conexión del cliente Android con los datos de la parafarmacia.

La documentación del sistema tendrá una comparación entre el proyecto fin de grado y el proyecto fin de máster, tanto de diseño, como de las herramientas utilizadas y los estándares web. Se tratará de resaltar los aspectos que se han mejorado entre una versión y otra para ver la evolución que ha supuesto cursar el Máster en Ingeniería Web.

Capítulo 2. Introducción

2.1 Estudio de la Situación Actual

Para el proyecto se partirá, inicialmente, del trabajo realizado para finalizar los estudios en el Grado en Ingeniería Informática, que ya define en gran medida lo que se va a desarrollar en este proyecto.

A nivel de requisitos, se mantienen la mayoría de ellos, el cambio significativo vendrá con el uso de nuevas tecnologías para el desarrollo de parte del proyecto y con el uso de tecnologías ya utilizadas, pero de una forma más profesional, con todo lo que se ha aprendido durante el máster.

Además, se ha planeado realizar una documentación final que permita comparar ambos proyectos y ver de forma clara cuales han sido las mejoras introducidas y el resto de cambios.

2.2 Evaluación de Alternativas

A continuación, vamos a ver las diferentes opciones que se han valorado a la hora de realizar el análisis del proyecto.

2.2.1 Parafarmacia Web

Gestores de contenido

Para desarrollar la parafarmacia web se ha valorado el uso de un gestor de contenidos debido a las facilidades que aportan.

Alguna de las ventajas que aportan los gestores de contenido son las siguientes:

- La compatibilidad con múltiples pasarelas de pago.
- Tiene un potente control del catálogo para llevar la gestión de los productos. Esta ventaja, con el desarrollo del cliente Android, no se utilizaría.
- Su panel de control es muy sencillo de utilizar, accesible e intuitivo.
- Se pueden añadir más funciones a través de diferentes módulos.
- Utiliza seguridad SSL, para proteger los pagos de los clientes.
- Facilidad para traducir los textos a otros idiomas.
- Buena gestión de clientes y pedidos, ya que es un gestor enfocado al desarrollo de tiendas online.

El problema por el que se descarta este sistema es porque se quiere desarrollar una aplicación en la cual se pueda controlar todo el código desarrollado. Los gestores de contenidos ya facilitan en gran medida la tarea de generar una aplicación web construyendo la estructura base y permitiendo personalizarla mediante plugins, por esta razón, no tenía mucho sentido desarrollar un generador para algo similar a un generador.

Spring

Spring es un framework que se basa en el uso de ficheros XML y anotaciones para la creación de los objetos de la aplicación, así como su integración. Es de código abierto y utiliza Java como lenguaje de programación. La versión actual es la 5.0, pero para el proyecto se utilizará una de las versiones anteriores, la 3.1.

Se ha escogido Spring MVC debido a que se centra en facilitar la construcción de aplicaciones web sobre una plataforma muy conocida como es Java. Después de ver los EJB en el grado, para el máster se nos ha ofrecido esta solución como alternativa y sustituto para ellos.

Algunas de las principales ventajas que tiene el uso de Spring para el desarrollo de aplicaciones web son las siguientes:

- Se acopla a la aplicación sin obligar a modificar el código.
- Permite la inyección de dependencias, lo que facilita la comunicación de dos o más clases del proyecto.
- Añade la programación orientada a aspectos, que consiste en añadir funcionalidad extra a nuestras clases sin modificar su código.
- Permite la separación entre componentes de la aplicación, la parte del *front* y la parte del *back*.
- Utiliza lenguajes comunes de las aplicaciones web de forma simple.

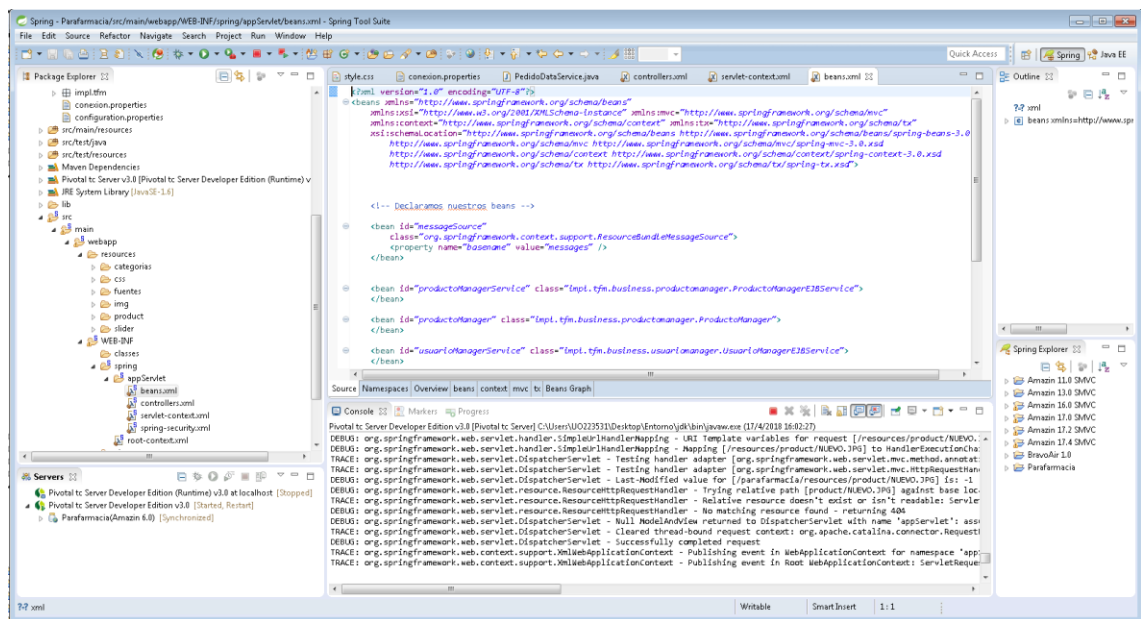


Ilustración 1 Ejemplo del uso de beans en Spring

2.2.2 Cliente Android

PhoneGap

PhoneGap es un *framework* para desarrollar aplicaciones móviles, permite crear, en teoría aplicaciones para diferentes sistemas operativos utilizando un único código. Este framework utiliza lenguajes de programación típicos de las aplicaciones web, como son HTML, CSS o JavaScript.

El problema por el que se ha descartado su uso es que, al no ser aplicaciones nativas para un solo sistema operativo, no permiten usar todo el potencial de cada uno de ellos, además, el cliente para gestionar la parafarmacia web se quiere únicamente para el sistema operativo Android, por lo que es mejor utilizar un entorno de trabajo más cercano y que permite utilizar todas las características de dicho sistema.

Android Studio

Android Studio es un entorno para desarrollar aplicaciones móviles para el sistema operativo Android. Está basado en IntelliJ IDEA, ya que posee sus herramientas de desarrollo y su potente editor de código. Su sistema de compilación se basa en *Gradle*, tiene un emulador para probar las aplicaciones, permite aplicar cambios en el desarrollo de forma rápida sin necesidad de compilar un nuevo APK y tiene herramientas que permiten detectar problemas de rendimiento, usabilidad y compatibilidad de versiones entre otras cosas.

Se ha decidido utilizar este entorno ya que se dispone de un conocimiento más amplio y se centra en el desarrollo de aplicaciones para dispositivos Android, por lo que al ser nativo de este sistema operativo, es más fácil usar o implementar las características propias.

2.2.3 Generador

Sirius

Sirius es una plataforma para Eclipse que permite crear DSLs (*Domain Specific Language*) gráficos en base a un modelo para la generación de código.

Sirius tiene la capacidad para crear herramientas como editores gráficos u otras que representen los datos en forma de diagramas, tablas o árboles.

La principal ventaja que tiene es que permite crear sin escribir código fuente y permite visualizar los modelos de manera rápida.

Se ha decidido utilizar XText para crear el DSL debido a que se conocía más en profundidad su funcionamiento y ya se tenía experiencia previa.

XText

Xtext es un *framework* para la creación de lenguajes de dominio específico creado por *Itemis*, utilizado por la plataforma Eclipse e IntelliJ entre otras. Es similar a Sirius, pero no permite visualizar los modelos de forma gráfica. Provee un editor de texto para Eclipse con su correspondiente analizador sintáctico basado en ANTLR (*ANOther Tool fo Language Recognition*)

Con esta herramienta podemos construir la gramática del lenguaje y crear con ella el modelo que permitirá generar automáticamente los diferentes artefactos.

Como ya se ha dicho en el apartado anterior, Xtext será el *framework* elegido para desarrollar el generador, ya que se ajusta mejor a las necesidades del proyecto y se conoce mejor su funcionamiento.

2.2.4 Bases de datos

Se conoce como base de datos a los bancos de información que contienen datos relativos a diversas temáticas, que comparten entre sí algún tipo de relación que los pueda ordenar o clasificar en conjunto. Estos datos son almacenados sistemáticamente para su posterior uso.

Las principales alternativas estudiadas para el proyecto son las siguientes:

MySQL

MySQL es un sistema de gestión de bases de datos relacionales desarrollado por Oracle Corporation, es de código abierto y se considera como la base de datos con mayor popularidad para las aplicaciones web.

Alguna de las ventajas que podemos tener al usar esta base de datos son, la velocidad al realizar las operaciones, ya que tiene un gran rendimiento, la facilidad para su creación, configuración e instalación, su bajo consumo permite la ejecución en máquinas con pocos recursos, tiene una mínima probabilidad para corromper los datos y, por último, su seguridad.

Para el desarrollo de la parafarmacia y los servicios web se utilizará esta base de datos por todas las ventajas descritas anteriormente.

HSQldb

HSQldb (*Hyper SQL Databases*) es un sistema de gestión de bases de datos relacionales escrito en JAVA que utiliza los *drivers* de JDBC (*Java Database Connectivity*).

Se utiliza para desarrollos con JAVA, sus principales ventajas son, la velocidad de arranque, la velocidad para realizar operaciones y el uso del lenguaje SQL estándar.

Esta base de datos está enfocada en el soporte de pruebas, ya que es mucho más simple que otras como MySQL.

SQLite

SQLite es un sistema de gestión de bases de datos relacionales, la principal diferencia con otras bases de datos es que pasa a formar parte integral del programa que lo utiliza, en el resto de casos, el programa y la base de datos se comunican, pero son partes independientes. Esta diferencia hace que el acceso y la comunicación sea mucho más eficiente ya que usa funciones internas.

Las aplicaciones Android utilizan SQLite como el principal gestor de bases de datos, por lo que será utilizada en el desarrollo del cliente Android.

MongoDB

El nombre de MongoDB viene de la palabra *humongous*, que significa enorme, es un sistema de base de datos NoSQL orientado a documentos. En lugar de almacenar los datos en tablas, los almacena en documentos con un esquema dinámico, lo que permite una integración más rápida y fácil para ciertas aplicaciones que requieran escalabilidad.

Las bases de datos NoSQL no tienen toda la potencia que ofrece un sistema relacional y no aporta ninguna característica destacable para nuestro proyecto.

Capítulo 3. Aspectos Teóricos

En este capítulo se hablará de los términos o aspectos más importantes que se van a utilizar y repetir durante el desarrollo del proyecto.

3.1 Lenguajes de Dominio Específico (DSL)

Un lenguaje de dominio específico es un lenguaje de programación que permite resolver un problema en concreto a partir de notaciones y abstracciones, para ello se define una sintaxis concreta a partir del dominio de un problema y a partir de ahí se genera un lenguaje especialmente pensado para resolver los problemas de ese dominio. Existen muchos lenguajes conocidos que son DSL (Domain-Specific Language), por ejemplo, SQL (Lenguaje de consultas de bases de datos).

Algunos de los beneficios de utilizar los DSL son:

1. Se expresa el dominio del problema en un nivel de abstracción.
2. Son concisos.
3. Son reutilizables para otros propósitos.
4. Aumentan la productividad.
5. Aumentan el mantenimiento.
6. Aumentan la portabilidad.

En el diseño de un DSL se pueden distinguir varios elementos, los conceptos utilizados para el modelado, la salida generada, el modelo y la variabilidad, ya que estos lenguajes son flexibles para añadir variaciones en cualquier momento.

Para construir un DSL hay que seguir una serie de pasos:

- En primer lugar, se realiza la definición del dominio.
- En segundo lugar, se define la terminología que se va a utilizar.
- En tercer lugar, se describen los conceptos del dominio.
- Por último, se describen las constantes y variables junto a sus dependencias.

En este sistema, se ha creado un lenguaje de dominio específico en el desarrollo del generador. El lenguaje creado es lo más simple posible para permitir a los usuarios que vayan a utilizarlo que no sea necesario disponer de mucho tiempo para aprender a escribirlo, lo que les permitirá centrarse en personalizar al máximo las nuevas parafarmacias que se generen.

3.2 Servicio Web SOAP

Un Servicio Web es una tecnología utilizada para intercambiar datos entre aplicaciones escritas en diferentes lenguajes de programación y ejecutadas sobre cualquier tipo de plataforma.

El principal objetivo de esta tecnología es conseguir la mayor interoperabilidad posible, ya que de esta forma se consigue adoptar unos estándares más abiertos.

Los encargados de la arquitectura y reglamentación de estos servicios son OASIS y el W3C, también mencionado en los anteriores apartados.

Junto con esta tecnología se relacionan una serie de estándares utilizados. Estos estándares son:

- **Webservices Protocol Stack:** es un estándar o protocolo que define como implementar y hacer que los servicios web interactúen entre sí.
- **XML:** es un estándar conocido internacionalmente que representa información estructurada en la web, permitiendo que dicha información sea almacenada, visualizada, transmitida o procesada por diferentes tipos de aplicaciones.
- **SOAP:** las siglas de SOAP vienen de *Simple Object Access Protocol*, como ya se ha dicho anteriormente, es un estándar o protocolo del Servicio Web que define la forma de comunicarse de diferentes procesos mediante el uso de datos XML. Actualmente, es uno de los protocolos más utilizado en los servicios web.
- **WSDL:** las siglas de WSDL vienen de *Web Services Description Language*, es una definición escrita en XML de la interfaz que provee un servicio web que indica cómo desarrollar una solicitud para dicho servicio.
- **UDDI:** las siglas de UDDI vienen de *Universal Description, Discovery and Integration*, es un protocolo basado en SOAP que define la forma de publicar y encontrar información sobre servicios web.
- **WS-Security:** es un protocolo de comunicación que aporta el medio para aplicar seguridad a los servicios web.
- **REST:** las siglas de REST vienen de *REpresentational State Transfer*, es otro protocolo que permite crear servicios web usados por clientes que entiendan HTTP.

Para este proyecto se va a utilizar un Servicio Web SOAP, que utilizan mensajes XML para comunicarse y WSDL para generar una interfaz como punto de entrada.

En nuestro proyecto, esta tecnología se utiliza para intercambiar datos entre el cliente Android, el generador y la parafarmacia web. El cliente Android necesita saber que usuarios están registrados en la parafarmacia web o que productos hay en ella, así como los pedidos que le llegan, por eso, es necesario utilizar el Servicio Web para obtener toda esa información y poder modificarla. El generador también necesita conectarse con el Servicio ya que tendrá la opción de crear productos, los cuales deberán ser insertados en la base de datos utilizada por la parafarmacia para que, al generarse una nueva, ésta ya disponga de los productos necesarios.

Capítulo 4. Gestión de proyecto

4.1 Planificación

4.1.1 Planificación inicial

A continuación, se muestra una tabla con las tareas planificadas inicialmente para el desarrollo del proyecto:

Nº	Tarea	Nivel	Descripción
1	Estudio de la viabilidad	1	Se realizará un estudio previo del mercado para verificar si es adecuado el desarrollo del proyecto y permitirá obtener beneficios
2	Identificar stakeholders	1	Se realizará una lista de todos los interesados en el proyecto
3	Obtención y documentación de requisitos	1	Se listará una a una toda la funcionalidad de cada una de las partes del proyecto
4	Reunión para la aprobación del proyecto	1	Reunión realizada con el director del proyecto para su aprobación
5	Planificación del proyecto	2	Se desarrollará una lista de tareas que se van a llevar a cabo para el desarrollo del proyecto y se estimará la duración de cada una de ellas
6	Memoria	2	Se preparará la documentación del proyecto
7	Gestión de presupuestos	2	A partir de las necesidades de la aplicación, se calculará los recursos necesarios y se planificará el coste que tendrá su desarrollo
8	Gestión de riesgos	2	Lista de todos los posibles problemas que pueden aparecer durante el desarrollo de la aplicación
9	Arquitectura general	2	Se modelarán las diferentes partes que forman el proyecto para obtener una arquitectura que permita unir todas ellas
10	Modelos del sistema	2	Se hará una descripción general del modelo común a todas las partes del proyecto
11	Plan de pruebas	2	Se describirá como se van a desarrollar las pruebas para cada una de las aplicaciones
12	Desarrollo y diseño de la aplicación parafarmacia	3	Se empieza con las diferentes tareas que componen la aplicación web
13	Diseño de la arquitectura	3	Se construirá la arquitectura que se seguirá durante el desarrollo de la aplicación
14	Diseño de la base de datos	3	Se realizará el modelado de las diferentes tablas que compondrán la base de datos
15	Diseño de la interfaz	3	Se hará un diseño previo de las pantallas de la aplicación
16	Crear la base de datos	3	Se comienza con la creación de la base de datos y sus tablas
17	Desarrollo de la aplicación	3	Se realiza toda la lógica y código de la aplicación web
18	Desarrollo de la interfaz	3	Se desarrolla la interfaz que se ha diseñado previamente

19	Realizar las pruebas	3	Se realizan un conjunto de pruebas para comprobar el correcto funcionamiento de la aplicación
20	Desarrollo y diseño de los servicios web	4	Se comienza con las diferentes tareas que componen los servicios web
21	Diseño de la arquitectura	4	Se construirá la arquitectura que van a seguir los servicios
22	Desarrollo de los servicios del servidor	4	Se comienza a desarrollar los servicios en la parte del servidor, que se conectarán internamente con la base de datos de la aplicación web
23	Desarrollo de los servicios del cliente	4	Se comienza el desarrollo de los servicios en la parte del cliente, los que se utilizarán para enlazar la aplicación web con el cliente Android
24	Prueba de los servicios	4	Se realiza el conjunto de pruebas que permitirán comprobar el correcto funcionamiento de los servicios
25	Desarrollo y diseño del cliente Android	5	Se comienza con las diferentes tareas que componen el cliente Android
26	Preparar el entorno de desarrollo	5	Se instalará y pondrá apunto el entorno en el que se va a desarrollar el cliente Android
27	Crear el modelo de datos	5	Se hará el modelado de datos que va a tener internamente el cliente
28	Crear la base de datos	5	Se creará la base de datos junto con las tablas necesarias
29	Desarrollo de la lógica del cliente Android	5	Se realizará toda la lógica y código interno que permitirá el correcto funcionamiento del cliente Android
30	Integrar los servicios web	5	Se incluirán los servicios web que permiten obtener los datos de la aplicación web
31	Diseño de pantallas	5	Se hará un diseño e implementación de las pantallas
32	Realizar pruebas	5	Conjunto de pruebas que permitirán comprobar el correcto funcionamiento
33	Preparar servidor web	6	Se comienzan las tareas para la puesta a punto del servidor web
34	Instalar SO del servidor	6	Se instalar el sistema operativo que tendrá el servidor
35	Configurar la seguridad	6	Se configura la seguridad del sistema operativo
36	Instalar el servidor	6	Se realiza la instalación del servidor
37	Añadir seguridad al servidor	6	Se aplican los métodos necesarios para añadir seguridad al servidor y que sea objetivo del menor número de ataques posible
38	Desplegar aplicación web	6	Se realiza el despliegue de la aplicación web en el servidor
39	Realizar pruebas de integración y funcionamiento de la aplicación web y el cliente Android	7	Se realizan las pruebas conjuntas de todas las partes del proyecto para comprobar que todo se integra correctamente y funciona como se espera
40	Reunión para la aceptación y cierre del proyecto	7	Se realiza una reunión con el director del proyecto para su aceptación y posterior cierre
41	Entrega del proyecto	7	Se realiza la entrega del proyecto

Tabla 1 Tareas planificadas

4.1.2 Planificación inicial detallada

El proyecto se ha estimado en 490 horas de trabajo, siendo la fecha de inicio el 8 de noviembre de 2017 y la fecha de finalización el 27 de abril de 2018. Cuenta con varios hitos intermedios, en primer lugar, la reunión inicial para la aprobación del proyecto. Para dicha reunión se debe entregar la primera parte de la documentación que contiene el resumen, objetivos y alcance. En segundo lugar, al ser un proyecto claramente dividido en tres partes, parafarmacia web, cliente Android y generador, se ha considerado incluir un hito para cada una de las partes en el que se enseñará su funcionamiento. Por último, se ha puesto un hito final que consiste en la revisión y entregar de todo el proyecto.

▲ Análisis	4 días	lun 20/11/17	lun 11/12/17
Estudio de la viabilidad	1 día	lun 20/11/17	jue 23/11/17
Identificar stakeholders	6 horas	vie 24/11/17	mar 28/11/17
Obtención y documentación de requisitos	2 días	mié 29/11/17	vie 08/12/17
Reunión para la aprobación del proyecto	2 horas	lun 11/12/17	lun 11/12/17

Ilustración 2 Tareas del análisis

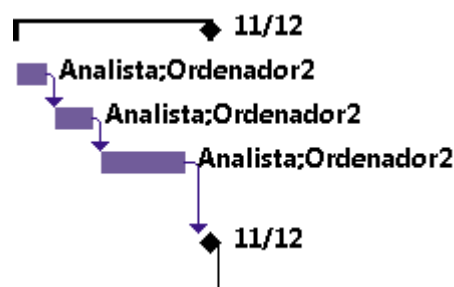


Ilustración 3 Diagrama de Gantt del análisis

Documentación	6,63 días	mar 12/12/17	mié 17/01/18
Planificación del proyecto	1 día	mar 12/12/17	vie 15/12/17
Memoria	1 día	lun 18/12/17	jue 21/12/17
Gestión de presupuesto	11 horas	vie 22/12/17	vie 29/12/17
Gestión de riesgos	6 horas	vie 29/12/17	mié 03/01/18
Arquitectura general	8 horas	mié 03/01/18	mar 09/01/18
Modelos del sistema	6 horas	mar 09/01/18	vie 12/01/18
Plan de pruebas	4 horas	vie 12/01/18	mar 16/01/18
Validar la documentación	2 horas	mar 16/01/18	mié 17/01/18

Ilustración 4 Tareas de documentación

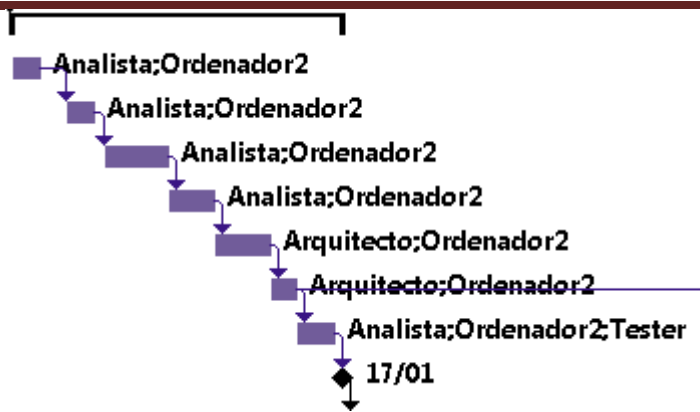


Ilustración 5 Diagrama de Gantt de la documentación

Desarrollo y diseño	16,88 días	mié 17/01/18	vie 20/04/18
▲ Aplicación parafarmacia	5,13 días	mié 17/01/18	mié 14/02/18
Diseño arquitectura	2 horas	mié 17/01/18	jue 18/01/18
Diseño base de datos	2 horas	jue 18/01/18	vie 19/01/18
Diseño de interfaz	2 horas	vie 19/01/18	lun 22/01/18
Crear base de datos	2 horas	lun 22/01/18	mar 23/01/18
Desarrollo de la aplicación	15 horas	mar 23/01/18	jue 01/02/18
Desarrollo de la interfaz	12 horas	vie 02/02/18	vie 09/02/18
Realizar las pruebas	6 horas	lun 12/02/18	mié 14/02/18

Ilustración 6 Tareas de desarrollo de la parafarmacia

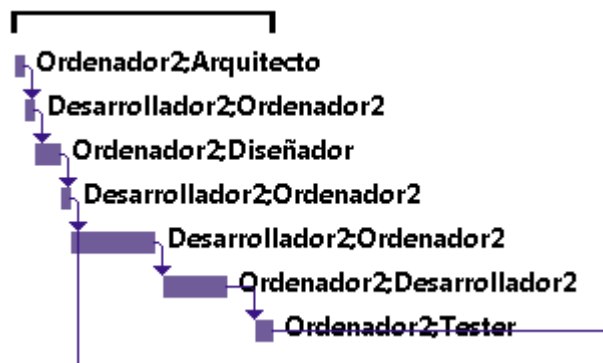


Ilustración 7 Diagrama de Gantt del desarrollo de la parafarmacia

Servicios web	1,38 días	mar 23/01/18	mar 30/01/18
Diseño arquitectura	1 hora	mar 23/01/18	mar 23/01/18
Desarrollo de los servicios	7 horas	mié 24/01/18	lun 29/01/18
Pruebas de los servicios	3 horas	lun 29/01/18	mar 30/01/18

Ilustración 8 Tareas de desarrollo del servicio web

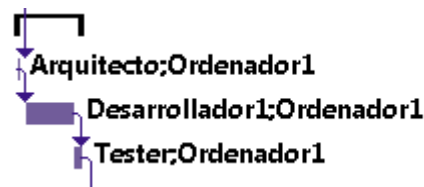


Ilustración 9 Diagrama de Gantt del desarrollo del servicio web

Cliente Android	14,5 días	mié 31/01/18	vie 20/04/18
Preparar entorno de desarrollo	1 hora	mié 31/01/18	mié 31/01/18
Crear base de datos SQLite	3 horas	mié 31/01/18	jue 01/02/18
Crear la tabla de la base de datos	1 hora	lun 19/03/18	lun 19/03/18
Desarrollo de la lógica de la aplicación	16 horas	mar 20/03/18	jue 29/03/18
Integrar servicios web para obtener los datos de la parafarmacia	10 horas	vie 30/03/18	jue 05/04/18
Diseño de las pantallas de la aplicación	16 horas	vie 06/04/18	mar 17/04/18
Realizar las pruebas	6 horas	mié 18/04/18	vie 20/04/18

Ilustración 10 Tareas de desarrollo del cliente Android

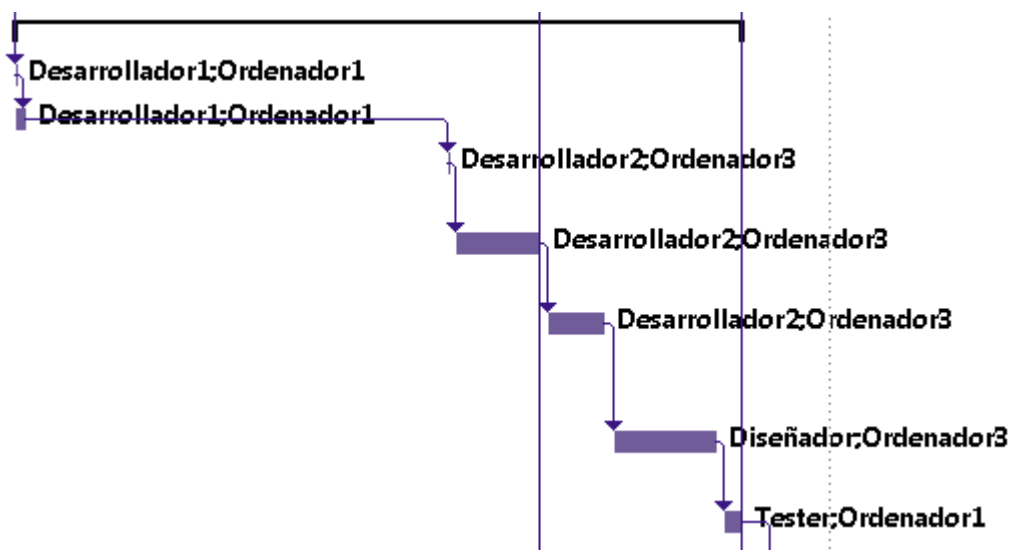


Ilustración 11 Diagrama de Gantt del desarrollo del cliente Android

Servidor Web	4,25 días	jue 29/03/18	vie 20/04/18
Instalar SO que soportará el servidor	2 horas	jue 29/03/18	jue 29/03/18
Añadir seguridad al equipo	2 horas	jue 05/04/18	jue 05/04/18
Instalar servidor web	8 horas	vie 06/04/18	mié 11/04/18
Añadir seguridad al servidor	10 horas	jue 12/04/18	mié 18/04/18
Desplegar parafarmacia	4 horas	jue 19/04/18	vie 20/04/18

Ilustración 12 Tareas de la preparación del servidor

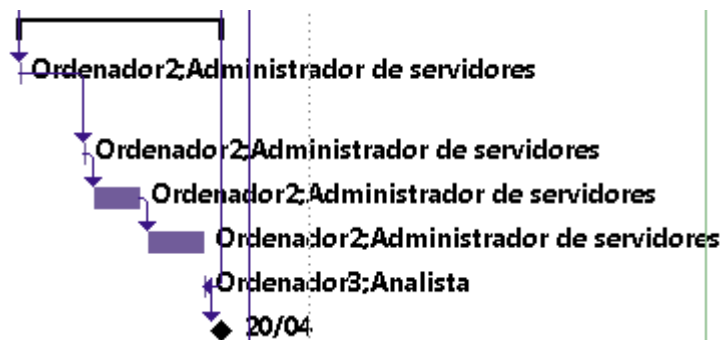


Ilustración 13 Diagrama de Gantt de la preparación del servidor

Realizar pruebas de integración y funcionamiento entre las aplicaciones	6 horas	lun 23/04/18	mié 25/04/18
Reunión para la aceptación y cierre del proyecto	2 horas	jue 26/04/18	jue 26/04/18
Entrega del proyecto	2 horas	vie 27/04/18	vie 27/04/18

Ilustración 14 Tareas de la fase de cierre del proyecto

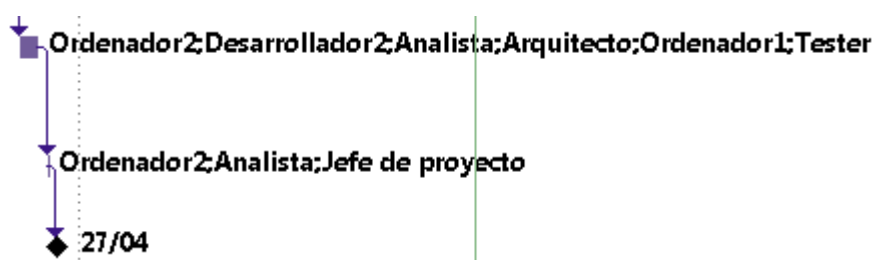


Ilustración 15 Diagrama de Gantt de la fase de cierre del proyecto

4.1.3 Problemas encontrados

Durante el desarrollo del proyecto se han encontrado una serie de problemas que se describen en el apartado 7.4.1 de este documento. Aquí no hablaremos de ellos ya que no han sido la causa por la que se ha retrasado la entrega del proyecto.

Las principales causas que han provocado el retraso en la entrega han sido, la mala planificación de las horas que iba a llevar realizar la documentación del sistema junto a no haber podido dedicar todas las horas semanales que se habían indicado en la planificación por diferentes circunstancias.

Pese a no haber cumplido con el objetivo inicial, se ha conseguido llegar a la siguiente entrega. Esta posibilidad ya se había valorado antes de iniciar el desarrollo por lo que no ha supuesto ningún problema, incluso ha servido como ayuda para entregar un producto de mayor calidad y más completo.

4.1.4 Planificación final real

La planificación final del proyecto asciende a un total de 530 horas de trabajo, siendo el día de inicio el 8 de noviembre de 2017, como se había planificado y la fecha final el 14 de junio de 2018, lo que quiere decir que se ha retrasado alrededor de un mes y medio.

La nueva planificación final solo cambia en uno de sus hitos, la documentación, que ha aumentado el tiempo de trabajo de 6,63 días a 8,5 días.

Documentación	8,5 días	jue 30/11/17	mié 17/01/18
Planificación del proyecto	1 día	jue 30/11/17	mié 06/12/17
Memoria	1 día	mié 06/12/17	mar 12/12/17
Gestión de presupuesto	9 horas	mar 12/12/17	lun 18/12/17
Gestión de riesgos	5 horas	mar 19/12/17	jue 21/12/17
Arquitectura general	11 horas	jue 21/12/17	jue 28/12/17
Modelos del sistema	11 horas	vie 29/12/17	vie 05/01/18
Plan de pruebas	11 horas	vie 05/01/18	vie 12/01/18
Validar la documentación	5 horas	lun 15/01/18	mié 17/01/18

Ilustración 16 Tareas de documentación reales

4.2 Presupuesto

4.2.1 Presupuesto resumido

El presupuesto mostrado a continuación se ha obtenido a partir de los costes de desarrollar cada una de las fases especificadas para el proyecto. La fase de análisis y viabilidad, que engloba el estudio previo al desarrollo, la fase de infraestructura, en la que se elabora la documentación y la arquitectura, la fase de diseño y desarrollo, donde se construyen las aplicaciones y la fase de documentación y aceptación que contiene información de todo lo que se ha realizado en el proyecto.

La suma de todas las fases nos da el presupuesto subtotal, sin sumarle el IVA, que asciende a 6.745,80 euros. El presupuesto final con la suma del IVA es de 8.162,42 €.

Fase	Coste
Fase de análisis y viabilidad	1.169,00 €
Fase de infraestructura	1.658,00 €
Fase de diseño y desarrollo	2.886,00 €
Fase de documentación y aceptación	923,00 €
Presupuesto antes de impuestos	6.745,80 €
IVA (21%)	1.416,618 €
Presupuesto con IVA	8.162,42 €

Tabla 2 Resumen del presupuesto

4.2.2 Presupuesto detallado

Para calcular el presupuesto he asignado diferentes roles que deberían estar en un proyecto real, en este caso serán todos realizados por el alumno a excepción del jefe de proyecto que será el director. Para cada rol se ha especificado el precio que costaría una hora de trabajo. Partiendo de la planificación anterior, se obtiene el número de horas que cada rol debe emplear para desarrollar el proyecto, así obtenemos el coste total por horas de trabajo:

Rol	Precio/Hora	Horas	Total
Desarrollador	30,00 €	74	2.220,00 €
Diseñador	25,00 €	4	100,00 €
Analista	35,00 €	57	1.995,00 €
Arquitecto	40,00 €	19	760,00 €
Tester	33,00 €	21	693,00 €
Administrador de servidores	38,00 €	14	532,00 €
Jefe de proyecto	48,00 €	7	336,00 €
Total roles		6.745,80 €	

Tabla 3 Coste roles por horas de trabajo

El proyecto incluye otros gastos relacionados con recursos materiales, estos son un servidor y un ordenador. No es necesario comprarlos así que se cuenta el gasto que van a tener por hora. Las horas se obtienen del número total de horas trabajadas, ya que siempre será necesario disponer de ambos recursos:

Material	Precio	Horas	Total
Servidor	0,3 €	183	54,9 €
Ordenador	0,3 €	183	54,9 €

Tabla 4 Coste material por horas de trabajo

A partir de los datos anteriores y la planificación realizada, se calcula el coste de desarrollar el proyecto para cada una de las siguientes fases: análisis, infraestructura, diseño, desarrollo y documentación. Cada una de las fases tiene asignados una serie de roles que participarán en ella con el número de horas planificadas. A partir del número de horas y el coste por hora se calcula el precio total de cada una de las fases.

Rol	Fase de análisis	Fase Infraestructura	Diseño, desarrollo	Documentación
Desarrollador	0	0	66	8
Diseñador	0	0	4	0
Analista	27	20	0	10
Arquitecto	2	14	0	3
Tester	0	6	10	5
Administrador de servidores	0	4	10	0
Jefe de proyecto	3	1	2	1
Precio/Fase	1.169,00 €	1.658,00 €	2.886,00 €	923,00 €
Total Fases	6.636,00 €			

Tabla 5 Coste total por fases

El presupuesto final del cliente lo obtenemos de la suma de todas las fases:

Fase	Coste
Fase de análisis y viabilidad	1.169,00 €
Fase de infraestructura	1.658,00 €
Fase de diseño y desarrollo	2.886,00 €
Fase de documentación y aceptación	923,00 €
Presupuesto antes de impuestos	6.745,80 €
IVA (21%)	1416,618
Presupuesto con IVA	8162,418

Tabla 6 Resumen final de costes

4.3 Gestión de riesgos

La metodología utilizada para realizar el plan de gestión de riesgos del proyecto será la descrita en el PMBOK (1).

Se tendrán en cuenta los diferentes apartados descritos a continuación:

Planificar la gestión de riesgos

Se tratará de crear el plan de gestión, que incluirá los criterios por los que se van a gestionar los riesgos, así como el registro y la priorización de los mismos, con esto se pretende aumentar las opciones de éxito que va a tener el proyecto.

Identificar los riesgos

En este proceso se determinarán todos los riesgos que estarán presentes durante el desarrollo del proyecto y se realizará su documentación incluyendo las características, la categoría, probabilidad de que aparezca, impacto que puede tener, descripción, las personas implicadas en el riesgo y las posibles formas de evitarlo y gestionarlo.

Realizar el análisis cualitativo de los riesgos

Con el objetivo de evaluar el impacto de los riesgos en el proyecto y poder tomar acciones, este proceso tratará de ordenarlos por prioridad, para ello se definirán diferentes niveles de prioridad.

Realizar el análisis cuantitativo de los riesgos

En este apartado se va a analizar de forma numérica el impacto que tendrían los riesgos que hemos identificado sobre cada uno de los objetivos finales del proyecto.

Planificar la respuesta a los riesgos

En este proceso vamos a intentar desarrollar una serie de opciones y acciones que se realizarán en caso de producirse cada riesgo para reducir al máximo el impacto de los riesgos sobre los objetivos del proyecto.

Implementar la respuesta a los riesgos

Este proceso tratará de ejecutar e implementar las respuestas descritas en el proceso anterior.

Monitorizar y controlar los riesgos

En este apartado se va a realizar el rastreo de los riesgos identificados anteriormente para controlar si ocurren dentro de nuestro proyecto y que acción final se realiza para tratar de evitarlo, además se podrán identificar nuevos riesgos que no se tuvieron en cuenta en la fase de identificación.

4.3.1 Riesgos identificados

A continuación, se muestra una tabla con los riesgos identificados para el proyecto junto la probabilidad de que ocurra, el impacto y la respuesta en caso de producirse.

El impacto del riesgo se calcula mediante la siguiente matriz de probabilidad e impacto:

Probabilidad	Muy Alta	0,90	0,05	0,14	0,27	0,50	0,81
	Alta	0,70	0,04	0,11	0,21	0,39	0,63
	Media	0,50	0,03	0,08	0,15	0,28	0,45
	Baja	0,30	0,02	0,05	0,09	0,17	0,27
	Muy Baja	0,10	0,01	0,02	0,03	0,06	0,09
			0,05	0,15	0,30	0,55	0,90
			Inapreciable	Bajo	Medio	Alto	Crítico
			0,05	0,15	0,30	0,55	0,90
			Impacto				

Nombre del riesgo	Probabilidad	Impacto	Respuesta
Riesgo porque el proyecto es más complicado de lo esperado	Alta	0,63	Evitar: Se tratará de planificar el proyecto junto a las tecnologías y arquitectura necesarias para intentar evitar que el proyecto se salga de lo planificado.
Riesgo de cambios en los requisitos del proyecto	Baja	0,17	Transferir: Se realizará un contrato para penalizar al responsable de los cambios durante el proyecto
Riesgo por falta de requisitos o por mal entendimiento de los mismos al realizar la obtención	Media	0,28	Evitar: Tras la obtención de los requisitos se realizará una reunión con el cliente para verificar que todos son correctos y no falta ninguno
Riesgo de que alguna funcionalidad no sea compatible con las versiones de dispositivos utilizados para las pruebas	Alta	0,39	Mitigar: Se intentará disminuir la posibilidad de que ocurra el riesgo teniendo varios dispositivos de diferentes marcas y con distintas versiones
Riesgo por mala elección de las tecnologías utilizadas para el desarrollo	Media	0,15	Evitar: Se tratará de hacer un estudio previo y hablar con algún experto (director del proyecto) para seleccionar las tecnologías que mejor se ajusten a las necesidades del proyecto
Riesgo por falta de experiencia con el uso de las tecnologías utilizadas	Muy Alta	0,50	Aceptar: Al ser un proyecto fin de máster se aceptará que no se tenga suficiente experiencia con la tecnología elegida, ya que uno de los objetivos es el aprendizaje

Riesgo de que aparezcan errores que no se puedan solucionar con el conocimiento del desarrollador	Media	0,45	Aceptar: Se aceptará que el riesgo pueda ocurrir y se tratará de consultar con el director del proyecto posibles soluciones en función del tipo de error encontrado
Riesgo por la imposibilidad de integrar las aplicaciones desarrolladas	Media	0,28	Evitar: Se realizará una buena arquitectura del proyecto para intentar evitar que se de este riesgo
Riesgo porque el servidor deje de estar operativo	Alta	0,39	Mitigar: Si se produce este riesgo se podrá modificar el cronograma y trabajar en otras tareas que no dependan del servidor hasta que se solucione
Riesgo porque el hardware necesario para desarrollar el proyecto deje de funcionar	Alta	0,39	Aceptar: No se puede prevenir que el hardware se estropee por lo que se deberá aceptar y tratar de sustituirlo lo antes posible en caso de producirse
Riesgo por la pérdida de código desarrollado para el proyecto	Alta	0,39	Evitar: Se realizarán copias de seguridad semanales para evitar que la pérdida de código ocasione un gran retraso en el desarrollo
Riesgo de que el director del proyecto deje de estar disponible	Media	0,15	Aceptar: Este riesgo no se puede controlar, en caso de ocurrir se aceptará y se intentará continuar sin la supervisión proporcionada por el director
Riesgo por el que la comisión encargada de la aceptación de proyectos rechace el mismo	Baja	0,27	Evitar: Se intentará evitar que este riesgo suceda realizando una buena presentación del proyecto y revisándola con el director
Riesgo de producirse bugs al pasar la aplicación Web del entorno de desarrollo a producción	Media	0,28	Evitar: Se tratará de crear un entorno de desarrollo lo más parecido al de producción
Riesgo de no cumplirse con todas las leyes que influyen en el proyecto	Muy Alta	0,81	Mitigar: Mantenerse al tanto de las actualizaciones en las leyes que afectan al proyecto, pedir a un experto que revise si se cumple con todos los apartados necesarios y revisar otras Webs de comercio electrónico similares para tener una idea más acertada de lo que se debe hacer

Tabla 7 Riesgos del proyecto

Capítulo 5. Análisis

Este apartado contendrá toda la especificación de requisitos y toda la documentación del análisis de la aplicación, a partir de la cual se elaborará posteriormente el diseño.

5.1 Requisitos del Sistema

En este apartado se procederá a describir los requisitos que afectan a todas las partes del sistema, se dividirá en tres partes, una para cada subproyecto más general, la parafarmacia web, el cliente Android y el generador.

5.1.1 Parafarmacia Web

A continuación, se describen los requisitos de la parafarmacia web.

5.1.1.1 Cuenta de usuario

Código	Nombre Requisito	Descripción del Requisito
R1.1	Registro de usuario	El sistema debe permitir a los usuarios registrarse, para ello deberán introducir los siguientes datos obligatoriamente: <ul style="list-style-type: none"> • Correo electrónico. • Contraseña (por duplicado).
R1.2	Modificar usuario	El sistema debe permitir a un usuario registrado modificar sus datos personales.
R1.3	Iniciar sesión	El sistema debe permitir a un usuario ya registrado iniciar sesión, para ello deberá introducir el email y la contraseña.
R1.4	Cerrar sesión	El sistema debe permitir a un usuario cerrar sesión después de haber iniciado sesión.
R1.5	Ver pedidos	El sistema debe permitir al usuario ver una lista de los pedidos que ha realizado. Se mostrarán los siguientes datos sobre ellos: <ul style="list-style-type: none"> • Precio total. • Lista de productos. • Fecha. • Estado del pedido.

Tabla 8 Requisitos cuenta de usuario parafarmacia web

5.1.1.2 Compra

Código	Nombre Requisito	Descripción del Requisito
R2.1	Mostrar catálogo de productos	El sistema debe permitir a los usuarios ver todos los productos disponibles para comprar según la categoría a la que pertenezcan. <ul style="list-style-type: none"> • Nombre • Imagen del producto • Descripción • Precio • Categoría
R2.2	Añadir producto al carrito	El sistema debe permitir a los usuarios añadir productos a su carrito de la compra.
R2.3	Eliminar producto del carrito	El sistema debe permitir a los usuarios eliminar productos del carrito.
R2.4	Ver carrito	El sistema debe permitir al usuario ver la lista de productos que ha añadido a su carrito.
R2.5	Precio del carrito	El sistema debe mostrar al usuario la suma total del precio de todos los productos del carrito y el subtotal de cada uno de ellos.
R2.6	Comprar	El sistema debe permitir a un usuario con sesión iniciada en el sistema que finalice su compra. El sistema permitirá seleccionar el método de pago contra reembolso o pago con Paypal y mostrará al usuario un resumen final en el que se muestre: <ul style="list-style-type: none"> • Precio total con gastos de envío. • Lista de productos. • Dirección a la que se va a enviar el pedido.
R2.7	Buscar producto	El sistema permitirá al usuario buscar un producto introduciendo alguna palabra de su nombre.

Tabla 9 Requisitos compra de productos parafarmacia web

5.1.1.3 No funcionales

Código	Nombre Requisito	Descripción del Requisito
R3.1	Disponibilidad	El sistema debe estar disponible las 24 horas del día y ser accesible por cualquier usuario con conexión a internet.
R3.2	Navegadores	El sistema debe permitir su ejecución en los navegadores Internet Explorer, Google Chrome y Mozilla Firefox.
R3.3	Responsive	El sistema debe tener un diseño que permita ser visualizado desde dispositivos móviles y tablets.
R3.4	Rendimiento	El sistema debe permitir el acceso simultáneo de al menos mil usuarios sin disminuir su tiempo de carga.
R3.5	Diseño	El sistema mostrará mensajes de error cuando se produzca algún fallo o se realice alguna acción no permitida.
R3.6	Ley Oficial de Protección de Datos (LOPD)	<p>Según la Ley Oficial de Protección de Datos (LOPD) y la Ley de Servicios de la Sociedad de la Información (LSSI) el sistema debe cumplir los siguientes requisitos:</p> <ul style="list-style-type: none"> • Se protegerán los datos personales para mantener la intimidad personal y familiar. • Se debe informar al contratante de forma gratuita de los diferentes medios de carácter técnico que aumenten los niveles de seguridad de la información. • Se debe obtener el consentimiento del usuario para el tratamiento de sus datos personales. • Se incluirá un Aviso Legal, en el que se incluirán los datos básicos del negocio (Artículo 10). <ul style="list-style-type: none"> a. Propietario del sitio (nombre o denominación social) b. Datos de contacto: domicilio, dirección de correo electrónico, teléfono o número de fax. c. Número de Identificación Fiscal. d. Si está registrado en el Registro mercantil indicar el número de tomo, libro, folio y hoja. e. Si se encuentra registrado en otro registro distinto público del Registro Mercantil, indicar el número de inscripción que le corresponda. f. Si se encuentra adherido a algún código de conducta, indicarlo y facilitar el acceso de forma electrónica a dicho código de conducta. g. Si ejerce una actividad regulada, deberá indicar: <ul style="list-style-type: none"> ❖ Datos del Colegio profesional y número de colegiado. ❖ Título académico oficial o profesional. ❖ Dónde se expidió y si procede, su homologación.

		<ul style="list-style-type: none">❖ Normas profesionales que debe seguir, indicando cómo poder acceder a dicha información.• Se debe facilitar a cualquier usuario el acceso al Aviso Legal (Artículo 10)
R3.6	Usabilidad	El sistema debe cumplir con el nivel de accesibilidad web AA.

Tabla 10 Requisitos no funcionales parafarmacia web

5.1.2 Generador

A continuación, se describen los requisitos del generador de parafarmacias.

5.1.2.1 Funcionales

Código	Nombre Requisito	Descripción del Requisito
R4.1	Generar parafarmacias	El sistema debe permitir personalizar los siguientes campos de la parafarmacia generada: <ul style="list-style-type: none"> • Nombre • Titular • Dirección • Tipo de menú • Apariencia (colores de letras y fondos, tamaño de letras, márgenes laterales) • Categorías de los productos • Productos • Método de pago
R4.2	Productos generados	El sistema permitirá crear nuevos productos para cada tienda generada.
R4.3	Templates	El sistema incluirá Templates con el lenguaje de generación de la parafarmacia para facilitar su creación.

Tabla 11 Requisitos funcionales generador

5.1.2.2 No funcionales

Código	Nombre Requisito	Descripción del Requisito
R5.1	Instalación	El sistema debe de poder instalarse en ordenadores con sistema operativo Windows 7 o Windows 10.
R5.2	Diseño	El sistema mostrará mensajes de error cuando se introduzca código erróneo.

Tabla 12 Requisitos no funcionales generador

5.1.3 Cliente Android

A continuación, se describen los requisitos del cliente Android.

5.1.3.1 Funcionales

Código	Nombre Requisito	Descripción del Requisito
R6.1	Ver usuarios	El sistema mostrará una lista con los nombre e emails de los usuarios registrados en la parafarmacia web.
R6.2	Ver pedidos	El sistema mostrará una lista con los pedidos realizados por los usuarios de la parafarmacia web. Los datos mostrados serán: <ul style="list-style-type: none"> • Fecha • Estado • Lista de productos y su precio • Dirección de envío • Nombre del comprador
R6.3	Actualizar pedido	El sistema debe permitir al usuario actualizar el estado de los pedidos. Los posibles estados disponibles serán: <ul style="list-style-type: none"> • Preparando • Enviado • Entregado
R6.4	Ver productos	El sistema mostrará una lista con los productos que contiene la parafarmacia web.
R6.5	Actualizar producto	El sistema debe permitir al usuario actualizar los campos de un producto.
R6.6	Añadir producto	El sistema debe permitir al usuario añadir un nuevo producto a la parafarmacia web.

Tabla 13 Requisitos funcionales cliente Android

5.1.3.2 No funcionales

Código	Nombre Requisito	Descripción del Requisito
R3.1	Disponibilidad	El sistema debe estar disponible las 24 horas del día y ser accesible por cualquier usuario con conexión a internet.
R3.2	Dispositivos	El sistema debe permitir su ejecución en dispositivos móviles y tablets con sistema operativo Android para la versión 4.4 o superior.
R3.4	Rendimiento	El sistema debe permitir el acceso simultáneo de al menos 3 usuarios sin que se produzcan conflictos en los datos.

R3.5	Diseño	El sistema mostrará mensajes de error cuando se intente realizar alguna acción que no esté completa y mostrará mensajes de éxito cuando se complete una acción.
------	--------	---

Tabla 14 Requisitos no funcionales cliente Android

5.1.4 Identificación de Actores del Sistema

Usuario anónimo: se considera usuario anónimo a todo usuario que utilice la parafarmacia web y no se registre en ella. Estos usuarios podrán hacer varias cosas, ver los productos, añadir los productos al carrito, pero no finalizar su compra, buscar productos y ver las páginas de información.

Usuario registrado: se considera usuario registrado al usuario que ha creado una nueva cuenta en la parafarmacia web y ha iniciado sesión con ella. Estos usuarios podrán, además de hacer las mismas cosas que un usuario anónimo, finalizar la compra de los productos añadidos al carrito, modificar sus datos, ver los pedidos que ya ha realizado.

Administrador: se considera usuario administrador al usuario que utilice el cliente Android. No será necesario registrarse ni iniciar sesión ya que la aplicación solo será distribuida a los usuarios administradores. El administrador podrá realizar la gestión de la parafarmacia web, añadiendo productos nuevos, actualizando los que ya hay, ver los pedidos y los usuarios.

Desarrollador: se considera usuario desarrollador al usuario que se encargará de escribir el lenguaje para generar nuevas parafarmacias web.

5.1.5 Especificación de Casos de Uso

5.1.5.1 Parafarmacia Web

Nombre del caso de uso
Registrarse
Descripción
Un usuario entra por primera vez en la página web de la parafarmacia y quiere registrarse, para ello, se dirige al apartado de registro y se le pide que introduzca el correo electrónico junto con una contraseña por duplicado, nombre, dirección, ciudad y código postal. El sistema comprueba que el usuario no existe dentro del sistema y procede a crearlo.

Nombre del caso de uso
Iniciar sesión
Descripción
Un usuario entra a la página web de la parafarmacia con el fin de realizar una compra, para ello necesita iniciar sesión con su cuenta. Como ya dispone de una cuenta en el sistema, se dirige al apartado de inicio de sesión e introduce su correo electrónico o nombre y contraseña en los campos correspondientes. El sistema comprueba que dichos datos son correctos y permite al usuario iniciar sesión.

Nombre del caso de uso
Finalizar sesión
Descripción
Un usuario acaba de realizar una compra en la web de la parafarmacia, como no quiere seguir comprando, decide cerrar sesión, para ello accede al apartado de su cuenta y selecciona la opción de finalizar sesión. El sistema elimina los datos del usuario de la sesión.

Nombre del caso de uso
Ver productos
Descripción
Un usuario a la página web de la parafarmacia para realizar una compra, para ello, quiere ver todos los productos que están a la venta, así que se dirige al apartado del menú de productos y se le muestra una lista con todos los productos disponibles junto con su imagen y el precio.

Nombre del caso de uso
Ver pedido
Descripción
Un usuario acaba de realizar una compra en la web de la parafarmacia y no sabe si ha comprado las unidades necesarias de un producto. Para ver el pedido se dirige a su cuenta dónde le aparecerá una lista con todos los pedidos que ha realizado junto a los productos que ha comprado.

Nombre del caso de uso
Modificar datos
Descripción
Un usuario quiere cambiar la contraseña de su cuenta junto con su dirección, ya que se ha mudado de casa. Para ello se dirige a su cuenta y selecciona la opción de editar cuenta. Ahí le aparecen los campos ya existentes con sus datos actuales que podrá modificar. Deberá escribir la contraseña por duplicado.

Nombre del caso de uso
Ver carrito
Descripción
Un usuario está realizando una compra en la web de la parafarmacia, para ello ha añadido varios productos a su carrito. Ahora quiere ver los productos que ha ido añadiendo y su precio total, para ello selecciona la opción del carrito que le lleva a la lista de productos que quería ver.

Nombre del caso de uso
Realizar compra
Descripción
Un usuario ya ha añadido al carrito todos los productos que quiere comprar. Para continuar con la compra inicia sesión con su cuenta y se dirige al carrito de la compra donde se encuentran todos los productos que quiere comprar junto con el precio final. Para continuar con la compra selecciona la opción de continuar con la compra. Una vez que ha realizado este proceso, finaliza la compra y el sistema almacenará el pedido.

Nombre del caso de uso
Añadir producto
Descripción
Un usuario quiere realizar una compra en la web de la parafarmacia, para ello necesita añadir los productos que desea comprar, así que se dirige al catálogo en el que se encuentran todos los productos y selecciona la opción de añadir al carrito que se encuentra junto a cada uno de los productos.

Nombre del caso de uso
Eliminar producto
Descripción
Un usuario acaba de llenar el carrito con varios productos que desea comprar, pero uno de ellos ha sido añadido por error y desea eliminarlo, para ello, se dirige a la opción del carrito donde se le muestra la lista con los productos y selecciona el botón para quitar unidades.

Nombre del caso de uso
Filtrar productos
Descripción
Un usuario quiere realizar una compra en la web de la parafarmacia, para ello se encuentra en el catálogo de la tienda, pero hay tantos productos que no encuentra el producto que quiere, sabe que es una crema, pero no la encuentra. Por esta razón, se dirige al buscador de la tienda e introduce el texto “crema”. El sistema selecciona los productos que contienen esa palabra en su nombre y se los muestra al usuario. El resultado final son 5 productos, por lo que el usuario encuentra lo que buscaba.

Nombre del caso de uso
Ver detalles de producto
Descripción
Un usuario quiere comprar un producto, pero no tiene claro cuál es el que necesite realmente dado que muchos tienen nombres parecidos, así que selecciona la opción de ver detalles que se encuentra junto a cada uno de los productos y ahí encuentra la descripción detallada que estaba buscando.

5.1.5.2 Cliente Android

Nombre del caso de uso
Actualizar producto
Descripción
El administrador de la tienda quiere cambiar el precio de un producto, para ello se dirige a la lista de los productos y selecciona el botón para actualizar. Introduce el nuevo precio y guarda los cambios.

Nombre del caso de uso
Ver pedidos
Descripción
El administrador de la tienda quiere ver si ha recibido nuevos pedidos, para ello se dirige a la opción de pedidos y comprueba si se ha añadido alguno nuevo a la lista.

Nombre del caso de uso
Actualizar pedido
Descripción
El administrador de una tienda online ha enviado un pedido correctamente a su cliente y éste lo ha recibido, por lo que quiere que dicho pedido sea notificado a entregado, para ello se dirige a la lista de los pedidos y selecciona la opción de entregar pedido.

Nombre del caso de uso
Ver productos
Descripción
El administrador de la tienda online quiere ver el catálogo de productos, para ello se dirige a la opción de productos en la que le aparece una lista con todos ellos, pudiendo ver su imagen, nombre y categoría.

Nombre del caso de uso
Añadir producto
Descripción
El administrador de una tienda online quiere añadir un nuevo producto a su catálogo, para ello se dirige a la lista de productos y escoge la opción de añadir. Introduce los datos del nuevo producto, nombre, descripción, precio, categoría y selecciona la opción de añadir.

Nombre del caso de uso
Ver usuarios
Descripción
El administrador de una tienda online quiere comprobar cuantos usuarios hay registrados en su tienda, para ello se dirige a la opción de usuarios dónde le aparece una lista con todos los usuarios y la cantidad total.

5.1.5.3 *Generador*

Nombre del caso de uso
Añadir productos
Descripción
El desarrollador quiere crear una parafarmacia con varios productos en ella, para hacerlo, escribe en el código varias etiquetas de producto con sus datos correspondientes, el nombre, descripción, precio, categoría y disponibilidad. Una vez escrito el código, lo guarda y automáticamente el generador inserta los productos en la base de datos.

Nombre del caso de uso
Generar parafarmacia
Descripción
El desarrollador quiere crear una nueva parafarmacia completa, para ello escribe el código comenzando con la etiqueta parafarmacia y a continuación el resto de campos, menú, apariencia, productos, usuario... Una vez escrito el código, se guarda y se crean automáticamente los archivos que formarán la parafarmacia. Además, estos archivos se guardarán y compilarán ya en la parafarmacia nueva.

5.2 Identificación de los Subsistemas en la Fase de Análisis

5.2.1 Descripción de los Subsistemas

Para pasar a la descripción de los subsistemas, en primer lugar, haré una división más general de las partes más importantes de la aplicación. Para cada una de estas partes, realizaré la descripción de sus propios subsistemas. Las partes son: parafarmacia web, cliente Android, generador y servicios web SOAP.

5.2.1.1 Parafarmacia Web

5.2.1.1.1 Base de datos

El subsistema base de datos contiene las clases que permiten conectarse con la base de datos MySQL. Estas clases son completamente independientes del resto del proyecto gracias al uso del patrón DAO. Tenemos una interfaz que permitirá al resto de subsistemas comunicarse con él y una implementación de la interfaz para cada acceso a la base de datos, en este caso solo tendremos un acceso.

Dentro de este subsistema se incluyen las clases que representan el modelo de la base de datos, los usuarios, los productos y los pedidos.

Por último, tendremos las clases que implementan el acceso a la base de datos, éstas tendrán los métodos encargados de obtener, crear, eliminar y actualizar los usuarios, productos y pedidos.

5.2.1.1.2 Controladores

El subsistema formado por los controladores engloba todas las clases encargadas de recibir y gestionar las peticiones de los usuarios. Para cada vista de la parafarmacia web tenemos un controlador encargado de servir dicha página. Se encargan, además, de guardar al usuario en sesión cuando inicia sesión, conectarse con la interfaz que gestiona el modelo para insertar o coger elementos de la base de datos, enviar datos a las vistas o controlar errores entre otras cosas.

5.2.1.1.3 Vistas

El subsistema que recoge las vistas contiene todos los archivos JSP que almacenan el código estructurado que se va a mostrar al usuario. Además, se incluye en el subsistema los archivos CSS que, conjuntamente con los JSPs, permiten mostrar los elementos de una forma visualmente atractiva. Este subsistema se comunica con el anterior, ya que envía y recibe datos de los controladores para mostrar toda la información que no es estática.

5.2.1.2 Cliente Android

5.2.1.2.1 Base de datos

Este subsistema contiene todas las clases encargadas de crear y gestionar la base de datos SQLite propia de Android. Este subsistema no utiliza el patrón DAO si no que tiene una clase Helper encargada de la creación y una clase para cada una de las tablas de la base de datos que se encargan de obtener, insertar, actualizar o borrar los datos. Las tablas son las mismas que tenemos en el primer subsistema, usuarios, pedidos y productos.

5.2.1.2.2 Activities

Los Activities son las clases encargadas de controlar las vistas del cliente Android, su función es similar a la de un controlador en la parafarmacia web. Estas Activities se encargan de utilizar las clases que se conectan con la base de datos para obtener la información necesaria que se mostrará al usuario en cada una de las vistas. Alguno de los Activities que componen este subsistema son, el SplashActivity, que se encarga de conectarse con los servicios web para traer los datos necesarios y guardarlos en la base de datos SQLite, o el ProductosActivity encargado de coger los productos de la base de datos y enviárselos a la vista correspondiente.

5.2.1.2.3 Adaptadores

El subsistema compuesto por los adaptadores se utiliza por los Activities cuando se quiere mostrar al usuario elementos más complejos de los que Android permite. Para mostrar una lista en la que aparezca una imagen, nombre, precio y categoría de un producto es necesario crear un adaptador, ya que el elemento básico de Android que sería una lista, solo nos permitiría mostrar un texto.

5.2.1.2.4 Layouts

Los Layouts forman el conjunto de vistas que tendrá el cliente Android de cara al usuario. Son instanciados por los Activities cuando el usuario navega por la aplicación y están compuestos por varios elementos visuales, botones, campos de texto o imágenes entre otros.

5.2.1.3 Generador

5.2.1.3.1 Modelo

Este subsistema, aunque es generado automáticamente por el dominio del modelo desarrollado con el propósito de ser el lenguaje de generación de parafarmacias, es muy importante ya que conforma el esqueleto del generador. Estas clases con el modelo se utilizarán por el siguiente subsistema que se va a describir, los generadores.

5.2.1.3.2 Generadores

Las clases que forman este subsistema son las encargadas de generar el código que dará forma a la parafarmacia. Está formado por tres clases, la encargada de conectarse con los servicios web e insertar productos en la base de datos de la parafarmacia web, la encargada de generar el código CSS de la parafarmacia web y la encargada de generar las vistas o JSP de la parafarmacia Web.

5.2.1.4 Servicios Web SOAP

5.2.1.4.1 Base de datos

El subsistema de base de datos de los servicios web contiene las clases encargadas de conectarse y gestionar los datos. Es un subsistema similar al utilizado por la parafarmacia web. Para ahorrar código duplicado se podría haber utilizado los servicios web desde la parafarmacia, pero para conseguir que sea más eficiente y reducir los tiempos de respuesta, se decidió hacer una conexión separada y solo usar este subsistema cuando sea necesario.

5.2.1.4.2 Servicios

Este subsistema contiene las clases que ofrecen los métodos del servicio web. Estos métodos se crean a partir de la interfaz que ofrece el subsistema anterior para conectarse con la base de datos.

5.2.2 Descripción de los Interfaces entre Subsistemas

5.2.2.1 Parafarmacia Web

En la parafarmacia web se habían identificado tres subsistemas, la base de datos, los controladores y las vistas. Las comunicaciones que hay entre ellos son, en primer lugar, la base de datos y los controladores, esta comunicación se realiza a través de una interfaz de operaciones interna que ofrece el subsistema de la base de datos. En segundo lugar, las vistas se comunican con los controladores, esta comunicación se realiza de manera directa entre la vista y su controlador.

5.2.2.2 Cliente Android

En esta parte del proyecto se habían identificado cuatro subsistemas, la base de datos que se comunica con los Activities, esta comunicación se realiza a través de una interfaz de operaciones interna que ofrece la base de datos, los Activities se comunican con los Layouts, esta conexión es directa entre cada vista y su controlador. Los Activities también se comunican con los adaptadores, ya que necesitan de ellos para crear las vistas personalizadas. La comunicación entre ellos se realiza internamente en la aplicación.

Por último, los Activities se comunican con el subsistema externo con los servicios web. Esta conexión se realiza mediante el uso del protocolo SOAP.

5.2.2.3 Generador

El subsistema generador que engloba varios generadores diferentes se comunica con los servicios web mediante el protocolo SOAP, con el fin de acceder a la base de datos de la parafarmacia y poder añadir los productos creados.

5.3 Análisis de Interfaces de Usuario

5.3.1 Descripción de la Interfaz

A continuación, se explicarán los aspectos más importantes de las interfaces diseñadas para los diferentes subproyectos, la parafarmacia web, el cliente Android y el generador.

5.3.1.1 Parafarmacia Web

En el siguiente boceto se recogen los elementos más importantes que se mantendrán en todas las pantallas que forman la parafarmacia web.

Como podemos ver, lo primero que llama la atención es el título de la página junto al logo, ambos elementos estarán fijos en esa posición para todas las pantallas, con ello se pretende que el usuario sepa en todo momento en qué página web está tratando de comprar y tenga presentes los aspectos más importantes de la marca.

Otro de los elementos más importantes en una tienda online es el buscador, ya que permitirá al usuario encontrar rápidamente el producto deseado evitando con ello que se canse y abandone la web. Este buscador se ha colocado justo debajo de los elementos más visibles de la web, el logo y el título, por lo que se incluye en el recorrido visual que el usuario hará de la página.

Siguiendo al buscador, se encuentran los elementos del menú, en este apartado se trata de mostrar al usuario la menor cantidad de información posible, por lo que las categorías de productos se encuentran ocultas hasta que el usuario decida que quiere desplegarlas. Lo mismo ocurre con el submenú superior, cuando hemos iniciado sesión, tendremos la opción de desplegarlo para ver las opciones disponibles sobre la cuenta de usuario.

Justo debajo del menú, encontramos lo que se conoce como migas, se utilizan para guiar al usuario por la web y evitar que se pierda.

El resto de elementos de la web serán los que se irán modificando para mostrar al usuario el contenido que desee, en el boceto se muestra la forma en la que se ordenarán los productos, se puede ver que se siguen las mismas bases descritas anteriormente, mostrando únicamente lo que el usuario necesita ver y dando opciones para ver el resto de información, por ejemplo, los botones de descripción de los productos.

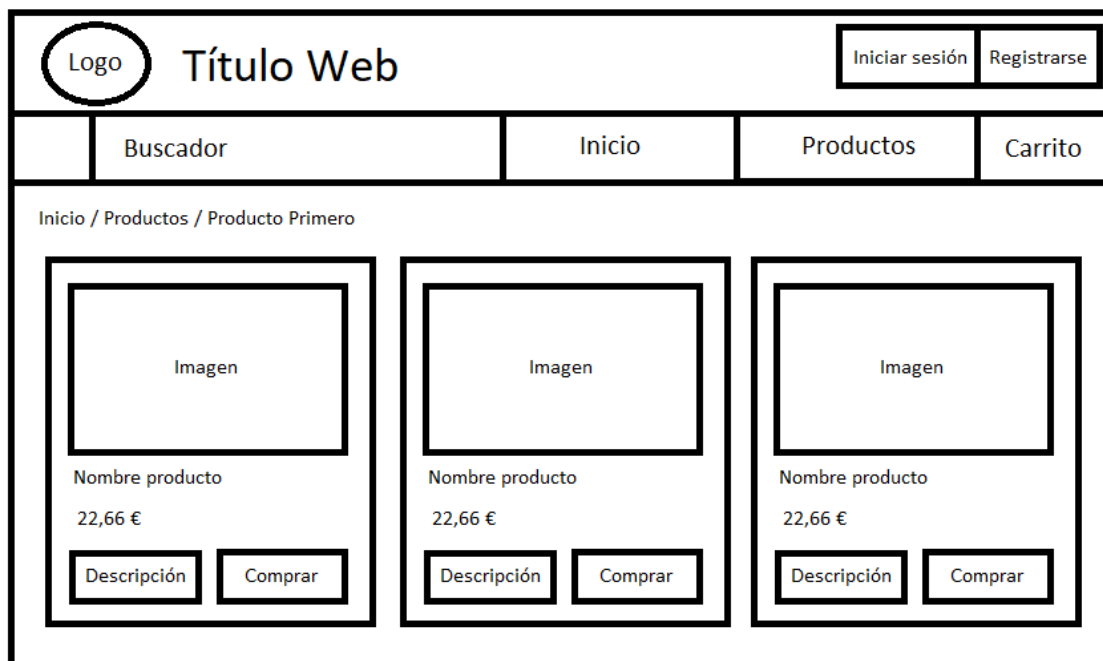


Ilustración 17 Boceto interfaz parafarmacia Web

5.3.1.2 Cliente Android

El diseño del cliente Android está enfocado a un usuario Administrador, por lo que no se van a tener en cuenta los mismos aspectos que en la parafarmacia web. En estas pantallas, en lugar de mostrar el nombre y logo en la parte superior, se va a mostrar algo más significativo que pueda ayudar al administrador a guiarse por la aplicación, por esta razón, a cada pantalla se le pondrá un título que haga referencia a la pantalla en la que se encuentra, “Productos”, “Añadir producto”, “Actualizar producto”.

Una de las cosas que mantenemos igual que en la parafarmacia web es la cantidad de información que se muestra y lo simples que van a ser las pantallas. Se quiere que cualquier persona sea capaz de administrar la tienda. Por ello, la página principal mostrará únicamente tres botones con las opciones más generales, además, se utilizará una guía en forma de colores para que el usuario sea capaz de reconocer y relacionar visualmente el apartado en el que se encuentra.

Por último, se ha tenido en cuenta la guía de diseño Material Design (1) de aplicaciones para dispositivos Android, por ejemplo, para asignar el tamaño de los botones, nunca inferior a 45dp, el tamaño de las letras, las combinaciones de los colores y el posicionamiento de los elementos, la colocación de las imágenes de los productos en la parte izquierda superior de cada uno de los elementos de la lista o los botones de volver a la pantalla anterior situados en la parte superior izquierda.

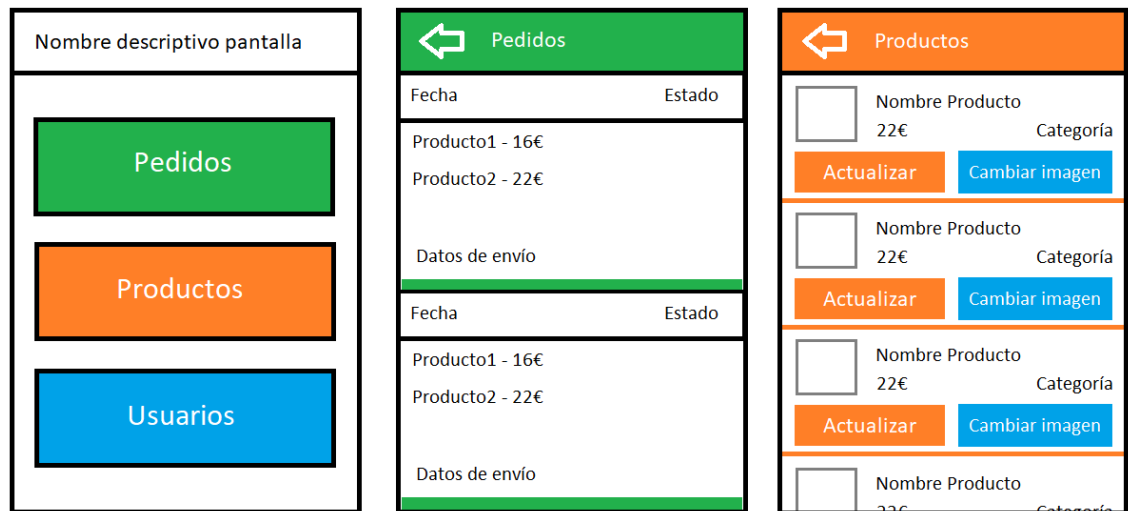


Ilustración 18 Boceto interfaz Cliente Android

5.3.1.3 Generador

El generador utiliza una interfaz propia que no permite apenas su personalización. Dicha interfaz es la equivalente de una ventana de *Eclipse*.

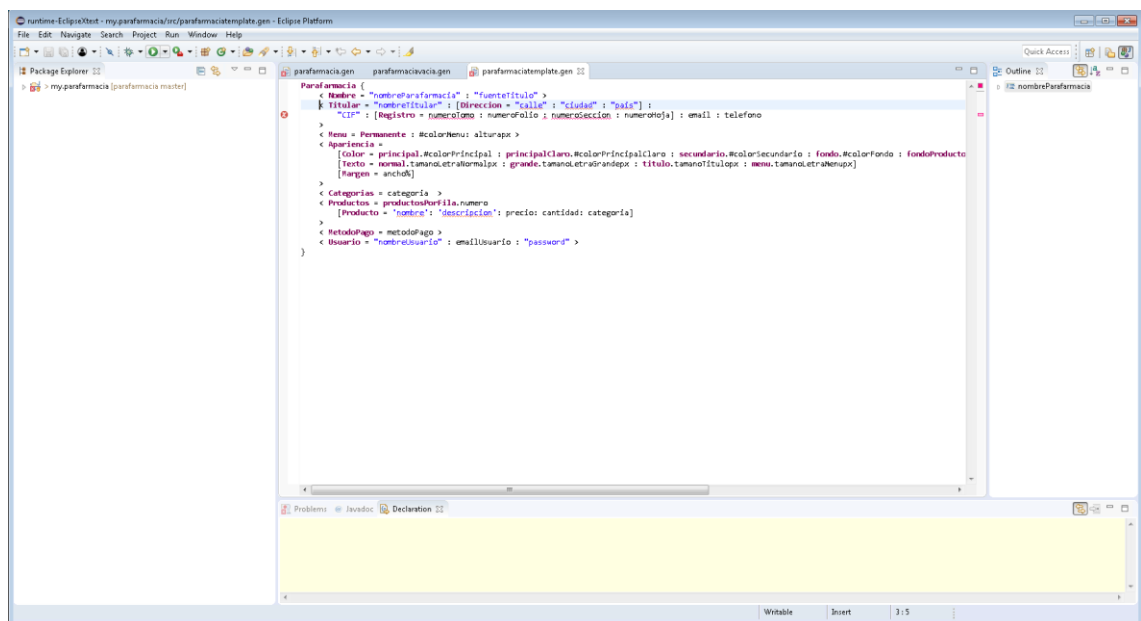


Ilustración 19 Interfaz generador

En esta interfaz podemos destacar algunas cosas, las palabras claves de lenguaje se pintan en color granate y negrita, lo que permite que destaquen por encima del resto, los campos personalizables se dividen en dos, los textos largos, que están en color azul y las palabras, números, identificadores o cualquier otro campo que se defina como un único elemento, se pinta en color negro.

Para señalar un error de sintaxis, aparece un icono rojo con una cruz blanca en la línea donde se ha cometido el error, además, pasando con el ratón por encima, se mostrará una descripción con el error.

Las ayudas para escribir el lenguaje se mostrarán al pulsar simultáneamente las teclas **Ctrl + Espacio**. Dicha ayuda consiste en un recuadro con varias opciones y recomendaciones para continuar escribiendo el código.

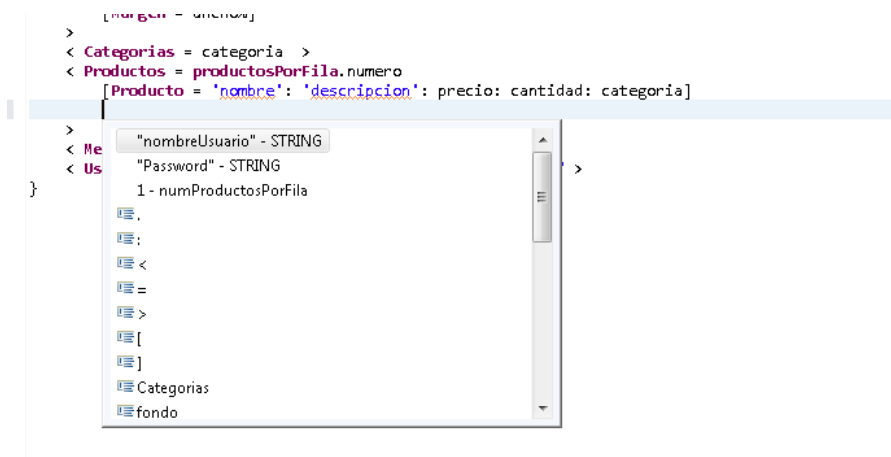


Ilustración 20 Ayuda para escribir el lenguaje del generador

Por último, se pueden ver más detalles generales de la interfaz como las pestañas de la parte superior que nos permiten abrir varios archivos de generación simultáneamente y navegar entre ellos, o el explorador que contiene los archivos generados y los archivos de generación, situado en la parte izquierda, que permitirá abrirlos pinchando sobre ellos.

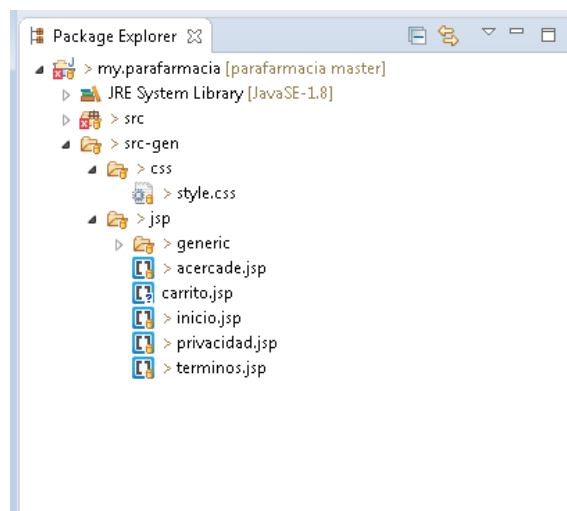


Ilustración 21 Explorador de archivos del generador

5.3.2 Diagrama de Navegabilidad

5.3.2.1 Parafarmacia Web

La siguiente imagen muestra el mapa de navegación de la parafarmacia web, en él se puede ver que se han diferenciado las pantallas que guardan alguna relación entre ellas. Las primeras que se ven contienen toda la información necesaria para cumplir con las leyes correspondientes, las de arriba a la derecha contienen el apartado de cuentas de usuario y su gestión, por último, las de abajo a la derecha tienen todo lo relacionado con la compra de productos.

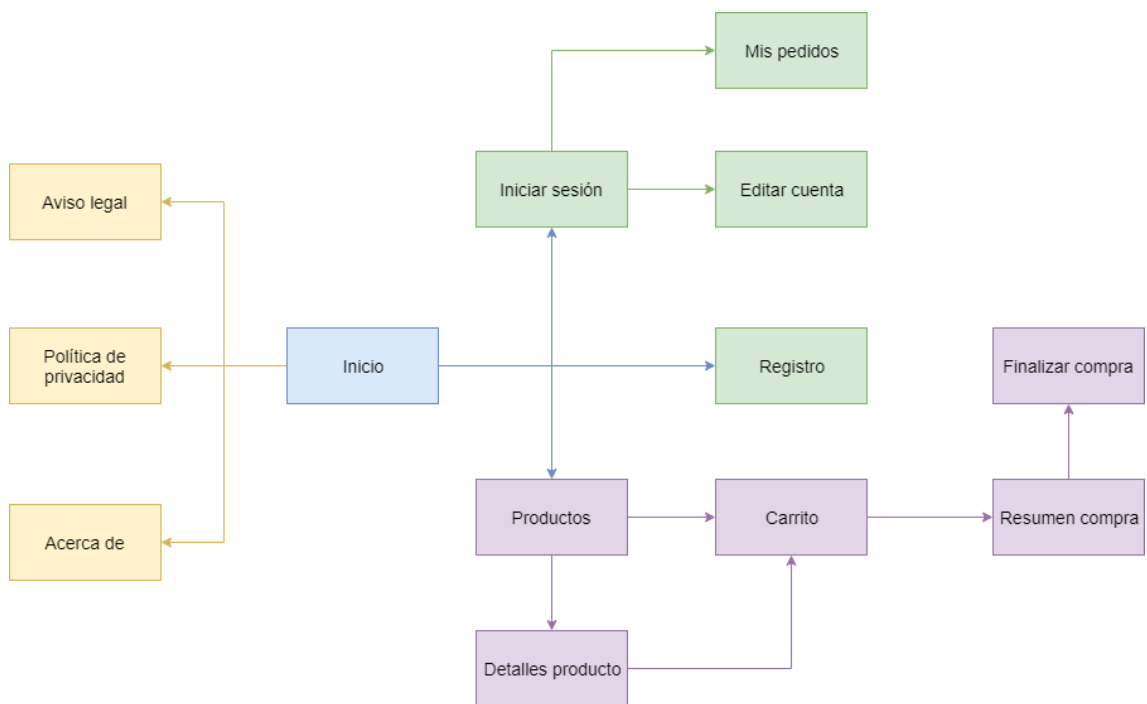


Ilustración 22 Diagrama de navegabilidad parafarmacia web

5.3.2.2 Cliente Android

La siguiente imagen contiene el mapa de pantallas del cliente Android, se puede ver que la navegabilidad es lo más simple posible, comienza por la pantalla inicial o splash, que dura 4 segundos y muestra el logo y nombre, a continuación, tenemos la ventana de inicio desde la que podemos ir a los usuarios, productos o pedidos. Por último, tenemos la opción de actualizar, editar la imagen o añadir un nuevo producto.

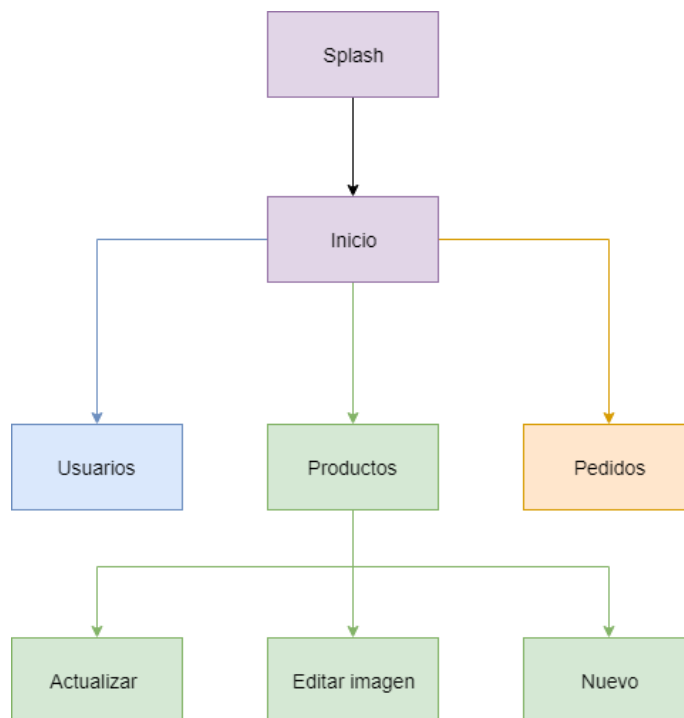


Ilustración 23 Diagrama de navegabilidad cliente Android

5.4 Especificación del Plan de Pruebas

En esta sección crearemos y diseñaremos el plan de pruebas de la aplicación y sus funciones, así como todos los mecanismos que utilizaremos para detectar errores y corregirlos ya en la fase de implementación.

5.4.1 Pruebas unitarias

Las pruebas unitarias realizadas serán generales para el subsistema parafarmacia web y el cliente Android, ya que ambos realizan las mismas operaciones con la base de datos, los resultados obtenidos deben ser también similares, por esta razón, se utilizan los mismos métodos y las mismas comprobaciones. El resto de partes no relacionadas con el acceso y modificación de la base de datos no tendrá pruebas unitarias, ya que sus métodos se encargan exclusivamente, tanto para la parafarmacia como para el cliente Android, de recibir las peticiones del usuario y servir la vista correspondiente enviando los datos obtenidos mediante los métodos de gestión de la base de datos.

Base de datos	
Prueba	Resultado esperado
Obtener usuarios: esta prueba consiste en realizar una consulta de obtención de todos los usuarios que contiene la base de datos y comprobar que el número de usuarios es correcto.	La consulta devolverá todos los usuarios de la base de datos.
Insertar usuario: esta prueba consiste en añadir un nuevo usuario a la base de datos y comprobar que se ha añadido correctamente.	La base de datos tendrá un nuevo usuario. Se comprobará que los datos del nuevo usuario coinciden con los datos insertados.
Actualizar usuario: esta prueba consiste en actualizar los datos de un usuario y comprobar que se han actualizado correctamente.	Se modificará cada uno de los campos que forman un usuario y se irá comprobando que se cambian al nuevo valor.
Obtener productos: esta prueba consiste en realizar una consulta de obtención de todos los usuarios que contiene la base de datos y comprobar que el número de productos es correcto.	La consulta devolverá todos los productos de la base de datos.
Insertar producto: esta prueba consiste en añadir un nuevo producto a la base de datos y comprobar que se ha añadido	La base de datos tendrá un nuevo producto. Se comprobará que los datos del nuevo producto coinciden con los datos insertados.

correctamente.	
Borrar producto: esta prueba consiste en borrar un producto de la base de datos y comprobar que la cantidad de productos se ha disminuido en una y el producto eliminado no está.	La base de datos tendrá producto menos.
Actualizar producto: esta prueba consiste en actualizar los datos de un producto y comprobar que se han actualizado correctamente.	Se modificará cada uno de los campos que forman un producto y se irá comprobando que se cambian al nuevo valor.
Insertar pedido: esta prueba consiste en añadir un nuevo pedido a la base de datos y comprobar que se ha añadido correctamente.	La base de datos tendrá un nuevo pedido. Se comprobará que los datos del nuevo producto coinciden con los datos insertados.
Actualizar pedido: esta prueba consiste en actualizar los datos de un pedido y comprobar que se han actualizado correctamente.	Se modificará cada uno de los campos que forman un pedido y se irá comprobando que se cambian al nuevo valor.
Obtener pedidos: esta prueba consiste en realizar una consulta de obtención de todos los pedidos que contiene la base de datos y comprobar que el número de pedidos es correcto.	La consulta devolverá todos los pedidos de la base de datos.

Tabla 15 Pruebas unitarias base de datos

Generador	
Prueba	Resultado esperado
Generar CSS: esta prueba consiste en escribir el DSL para generar una parafarmacia, incluyendo códigos de colores, márgenes y textos que se utilizarán para generar el CSS y comprobar que el archivo generado se corresponde con el esperado.	El archivo CSS generado tiene los nuevos códigos, márgenes y textos introducidos en el generador.
Generar JSP: esta prueba consiste en escribir el DSL para generar una parafarmacia, incluyendo elementos que formarán parte de los JSP, como las categorías, información de la tienda y del responsable.	Los archivos JSP generados tienen los elementos que se han introducido en el generador.

Tabla 16 Pruebas unitarias generador

5.4.2 Pruebas de integración y sistema

Estas pruebas serán muy importantes en un proyecto como éste, ya que las aplicaciones se encuentran conectadas entre sí.

Generador – Servicio Web	
Prueba	Resultado esperado
<p>Insertar productos: uno de los generadores que forman parte del generador, se encarga de limpiar la base de datos de la parafarmacia web e insertar los nuevos productos para dejarla nueva. Para conseguir esto, el generador hace uso de los servicios web, a los cuales, envía los datos necesarios para insertar nuevos productos. La prueba consiste en crear varios productos utilizando el método del generador y comprobar que los productos se insertan correctamente en la base de datos de la parafarmacia.</p>	<p>La base de datos borra sus datos antiguos y se insertan los nuevos productos.</p>

Tabla 17 Pruebas de integración y sistema generador - servicios web

Cliente Android – Servicio Web	
Prueba	Resultado esperado
<p>Nuevo producto: el cliente Android se conecta con el Servicio Web para crear nuevos productos en la parafarmacia web. La prueba consiste en insertar un nuevo producto.</p>	<p>La base de datos de la parafarmacia web tiene un nuevo producto con los datos que se han pasado desde el cliente Android.</p>
<p>Actualizar producto: esta prueba consiste en actualizar los datos de un producto.</p>	<p>Los nuevos datos del producto se han actualizado en la base de datos de la parafarmacia web.</p>
<p>Obtener productos: esta prueba consiste en solicitar todos los productos de la base de datos. Se comprobará tanto que el número de productos sea el correcto como los datos recibidos de cada producto.</p>	<p>El número de productos recibido es igual al de los productos que tiene la base de datos de la parafarmacia web y los datos de cada uno de los productos coincide con los de la base de datos.</p>
<p>Actualizar pedido: esta prueba consiste en actualizar el estado de un pedido, ya que es el único campo que se actualizará desde el</p>	<p>El estado del producto ha cambiado al nuevo estado que le pasamos.</p>

cliente Android.	
Obtener pedidos: esta prueba consiste en solicitar todos los pedidos de la base de datos. Se comprobará tanto que el número de pedidos sea el correcto como los datos recibidos de cada pedido.	El número de pedidos recibido es igual al de los pedidos que tiene la base de datos de la parafarmacia web y los datos de cada uno de los pedidos coincide con los de la base de datos.
Obtener usuarios: esta prueba consiste en solicitar todos los usuarios de la base de datos. Se comprobará tanto que el número de usuarios sea el correcto como los datos recibidos de cada usuario.	El número de usuarios recibido es igual al de los usuarios que tiene la base de datos de la parafarmacia web y los datos de cada uno de los usuarios coincide con los de la base de datos.

Tabla 18 Pruebas de integración y cliente Android - servicios web

5.4.3 Pruebas de usabilidad

Las pruebas de usabilidad que se van a desarrollar a continuación, se dividen en tres partes, pruebas para el cliente Android, pruebas para la parafarmacia web y pruebas para el generador de parafarmacias.

Las pruebas se realizarán sobre distintos tipos de usuarios: usuarios con conocimientos informáticos, usuarios que tengan relación con la tecnología y usuarios que no tengan conocimientos tecnológicos.

Consistirán en realizar diferentes recorridos por las pantallas de las aplicaciones para conseguir un determinado objetivo, de esta forma se puede medir el tiempo que tarda cada uno de los diferentes tipos de usuarios y comprobar que cosas se pueden mejorar para facilitar a los usuarios el uso de las mismas.

Se deberán completar una serie de cuestionarios con preguntas generales:

¿Usa un ordenador frecuentemente?
<ol style="list-style-type: none"> 1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca
¿Qué tipo de actividades realiza con el ordenador?
<ol style="list-style-type: none"> 1. Es parte de mi trabajo o profesión 2. Lo uso básicamente para ocio 3. Solo empleo aplicaciones estilo Office 4. Únicamente leo el correo y navego ocasionalmente

¿Ha usado alguna vez software como el de esta prueba?
<ol style="list-style-type: none"> 1. Sí, he empleado software similar 2. No, aunque si empleo otros programas que me ayudan a realizar tareas similares 3. No, nunca
¿Qué busca Vd. Principalmente en un programa?
<ol style="list-style-type: none"> 1. Que sea fácil de usar 2. Que sea intuitivo 3. Que sea rápido 4. Que tenga todas las funciones necesarias

Otras preguntas relacionadas directamente con el sistema:

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe dónde está dentro de la aplicación?</i>				
<i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i>				
<i>¿Le resulta sencillo el uso de la aplicación?</i>				
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. Espera?</i>				
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				
Calidad del Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
<i>El tipo y tamaño de letra es</i>				
<i>Los iconos e imágenes usados son</i>				
<i>Los colores empleados son</i>				

Diseño de la Interfaz	Si	No	A veces
¿Le resulta fácil de usar?			
¿El diseño de las pantallas es claro y atractivo?			
¿Cree que el programa está bien estructurado?			
Observaciones			
Cualquier comentario del usuario			

Tabla 19 Cuestionario pruebas usabilidad con usuarios

5.4.3.1 Cliente Android

Las pruebas sobre el cliente Android se realizarán sobre los usuarios comentados anteriormente.

Se pedirá a los usuarios que realicen las siguientes tareas:

- Ver los usuarios de la parafarmacia.
- Ver los pedidos de la parafarmacia.
- Actualizar el primer pedido que encuentre al estado entregado.
- Ver los productos de la parafarmacia.
- Actualizar el nombre de uno de los productos.
- Añadir un nuevo producto.

Para evaluar las tareas, se proporcionan una serie de preguntas para obtener información.

- *¿Qué tipo de usuario está realizando las tareas?*
- *¿El usuario comienza a realizar las tareas de forma inmediata?*
- *¿El usuario ha cometido algún error al realizar alguna tarea? ¿Cuál?*
- *¿Cuánto tiempo ha tardado el usuario en realizar cada una de las tareas?*

5.4.3.2 Parafarmacia Web

Las pruebas que se realizan sobre la parafarmacia web se realizan sobre los usuarios comentados anteriormente.

Se pedirá a los usuarios que realicen las siguientes tareas en el mismo orden que aparecen:

- Registrarse en la web.
- Cerrar sesión en la web.
- Iniciar sesión en la web.
- Ver la descripción de un producto.
- Realizar la compra de dos productos de la tienda.
- Visualizar la política de privacidad de la tienda.
- Finalizar la sesión.

Para evaluar las tareas se realizan una serie de preguntas:

- *¿Qué tipo de usuario está realizando las tareas?*
- *¿El usuario comienza a realizar las tareas de forma inmediata?*
- *¿El usuario ha cometido algún error al realizar alguna tarea? ¿Cuál?*
- *¿Cuánto tiempo ha tardado el usuario en realizar cada una de las tareas?*

5.4.3.3 Generador de parafarmacias

Para las pruebas sobre el generador, se realizarán sobre dos tipos de usuarios, los usuarios desarrolladores y usuarios que tengan conocimientos tecnológicos, ya que no está pensado su uso para usuarios sin conocimientos. Con estas pruebas se tratará de comprobar si el sistema y el lenguaje creado se puede utilizar por usuarios que no sean exclusivamente desarrolladores.

La prueba realizada para este apartado consiste en generar el código de una nueva parafarmacia. Se le pide que la nueva parafarmacia tenga tonos de colores verdes con un color de letra negro, un menú permanente que tenga 70px de altura, que se incluyan al menos dos categorías y un producto para cada una de ellas y que tenga los datos personales de la propia persona que está realizando la prueba.

Para obtener resultados se prepara un cuestionario:

- *¿El desarrollador comienza a escribir el código necesario de forma inmediata?*
- *¿En qué partes se han cometido errores y cuáles?*
- *¿Cuánto tiempo tarda el desarrollador en completar la tarea?*

5.4.3.4 Diseño adaptable

Uno de los aspectos importantes de la usabilidad es que el diseño se pueda adaptar a todos los tamaños de pantalla posibles, con esto se conseguirá que un usuario pueda realizar una compra en la parafarmacia desde un ordenador, una Tablet o un dispositivo móvil. Para comprobar que el diseño de la web es adaptable se utilizarán las siguientes herramientas:

Herramienta de Google:

https://search.google.com/test/mobile-friendly?utm_source=mft&utm_medium=redirect&utm_campaign=mft-redirect&hl=ES

Esta herramienta realiza un test automáticamente mediante el escaneo de la web y muestra el resultado indicando si es o no adaptable y los posibles fallos que pueda tener.

Quirktools:

<http://quirktools.com/screenfly/>

Esta herramienta permite cargar nuestra web y realizar diferentes acciones con ella. Las principales acciones que podemos hacer son, visualizar como se vería en los distintos dispositivos, ordenador, Tablet o móvil y visualizar la web sin imágenes ni CSS. La prueba principal que vamos a hacer con esta herramienta será mostrar la web para sus versiones móvil y Tablet para comprobar que se puede navegar por ella y los elementos se colocan adecuadamente.

5.4.4 Pruebas de accesibilidad

Para las pruebas de accesibilidad se van a utilizar varios métodos, uno proporcionado por dos herramientas online que realizan un test personalizado, TAW (<https://www.tawdis.net/#>) y WAVE (<http://wave.webaim.org/>). Por otro lado, se contestarán una serie de preguntas obtenidas de los estándares web desarrollados por la comunidad internacional World Wide Web (W3C), se rellenará otro formulario de verificación desarrollado por la misma comunidad, <https://www.w3.org/TR/WCAG10/full-checklist.html> y se harán diferentes tipos de pruebas con la página, como quitar las imágenes, aumentar el tamaño, quitar las hojas de estilos o aumentar el texto para comprobar que la página sigue permitiendo la navegación.

Cuestionario	
Pregunta	Respuesta
¿Los textos están escritos correctamente?	
¿Hay teclas de acceso rápido asignadas a los menús y a las funciones más utilizadas?	
¿Cada elemento no textual tiene un	

equivalente en forma de texto?	
¿Evita el uso de medidas absolutas?	
¿Está etiquetado cada control de los formularios y asociada la etiqueta con los controles?	
¿Hay mecanismos de navegación consistentes?	
¿Se evita que las páginas se refresquen automáticamente?	
¿Evita que se abran pop-ups o se cambie de ventana sin informar al usuario?	
¿Usa un lenguaje simple y claro?	
¿Identifica claramente el destino de cada enlace?	
¿La información transmitida a través de colores está disponible sin color?	
¿Hay suficiente contraste entre los colores de fondo y primer plano?	
¿Las páginas tienen un título corto y descripciones?	
¿Las URIs de los sitios son lo más cortas posibles?	
¿El contenido básico de la página se ve sin hacer scroll?	
¿Se evitan los enlaces externos?	
¿El tamaño global de la página es adecuado para las limitaciones de los dispositivos?	
¿Hay scroll en una única dirección?	
¿Se evitan las imágenes grandes o de alta resolución?	
¿Los formularios tienen valores por defecto?	

Tabla 20 Cuestionario pruebas de accesibilidad

5.4.5 Pruebas de rendimiento

Las pruebas de rendimiento en nuestro caso van a depender principalmente del servidor en el que se tenga alojada la parafarmacia web y los servicios web. También dependerá del número de accesos que tengamos a la base de datos, el número de conexiones a los servicios web, la carga de URL externas o la complejidad de las operaciones realizadas, además del número de usuarios que accedan al sistema simultáneamente.

Como es complicado poder medir el rendimiento de todo lo que se ha comentado, trataremos de tomar medidas generales de los tiempos de carga para las partes de la aplicación que más carga de trabajo puedan tener, como la página de productos en la parafarmacia web y en el cliente Android, ya que ambos deberán cargar imágenes y conectarse con la base de datos. También se medirán los tiempos que tarda el cliente Android en conectarse con los servicios web para realizar alguna modificación u obtención de datos.

Cuestionario	
Medir	Tiempo
Tiempo de carga de los productos en la parafarmacia web.	
Tiempo de carga de los productos en el cliente Android.	
Tiempo de carga de los pedidos en el cliente Android.	
Tiempo que tarda un producto en insertarse en la base de datos desde el cliente Android hasta la parafarmacia web.	
Tiempo que tarda el cliente Android en copiar los datos de la parafarmacia web a su base de datos.	
Tiempo que tarda el generador en crear una nueva parafarmacia.	
Tiempo que tarda la parafarmacia web en registrar un nuevo usuario.	

Tabla 21 Cuestionario pruebas de rendimiento

Capítulo 6. Diseño del Sistema

6.1 Arquitectura del Sistema

En la siguiente imagen se muestra una arquitectura general de todo el sistema, en ella se puede ver como parte principal la base de datos MySQL, que almacena los datos que van a ser utilizados por el resto de subsistemas. Para permitir que los otros subsistemas puedan utilizar la base de datos, aparece un nuevo elemento, los Servicios Web SOAP, que ofrecen métodos para conectarse con ella.

Podemos ver que la parafarmacia web no utiliza los Servicios Web para acceder a la base de datos si no que se conecta directamente, esto se ha diseñado así para conseguir un rendimiento mayor, ya que la parafarmacia web estará desplegada en el mismo servidor que su base de datos.

A la izquierda de la parafarmacia web nos encontramos el generador, este recibe un archivo con el lenguaje de creación de una parafarmacia y mediante los generadores se encarga de crear los archivos que componen la nueva parafarmacia web. Además, se conecta con los Servicios Web para limpiar la base de datos e insertar los nuevos productos, de esta manera, se reiniciará como si fuera nueva.

Por último, en la parte inferior se encuentra el Cliente Android, que tiene su propia base de datos SQLite en la que almacena una copia de los mismos datos que hay en la base de datos general. Esta copia será utilizada a modo de caché para que la aplicación funcione lo más rápido posible. El cliente Android, utiliza los Servicios Web, además de para copiar todos los datos, para insertar nuevos datos y de esta forma poder gestionar la parafarmacia web.

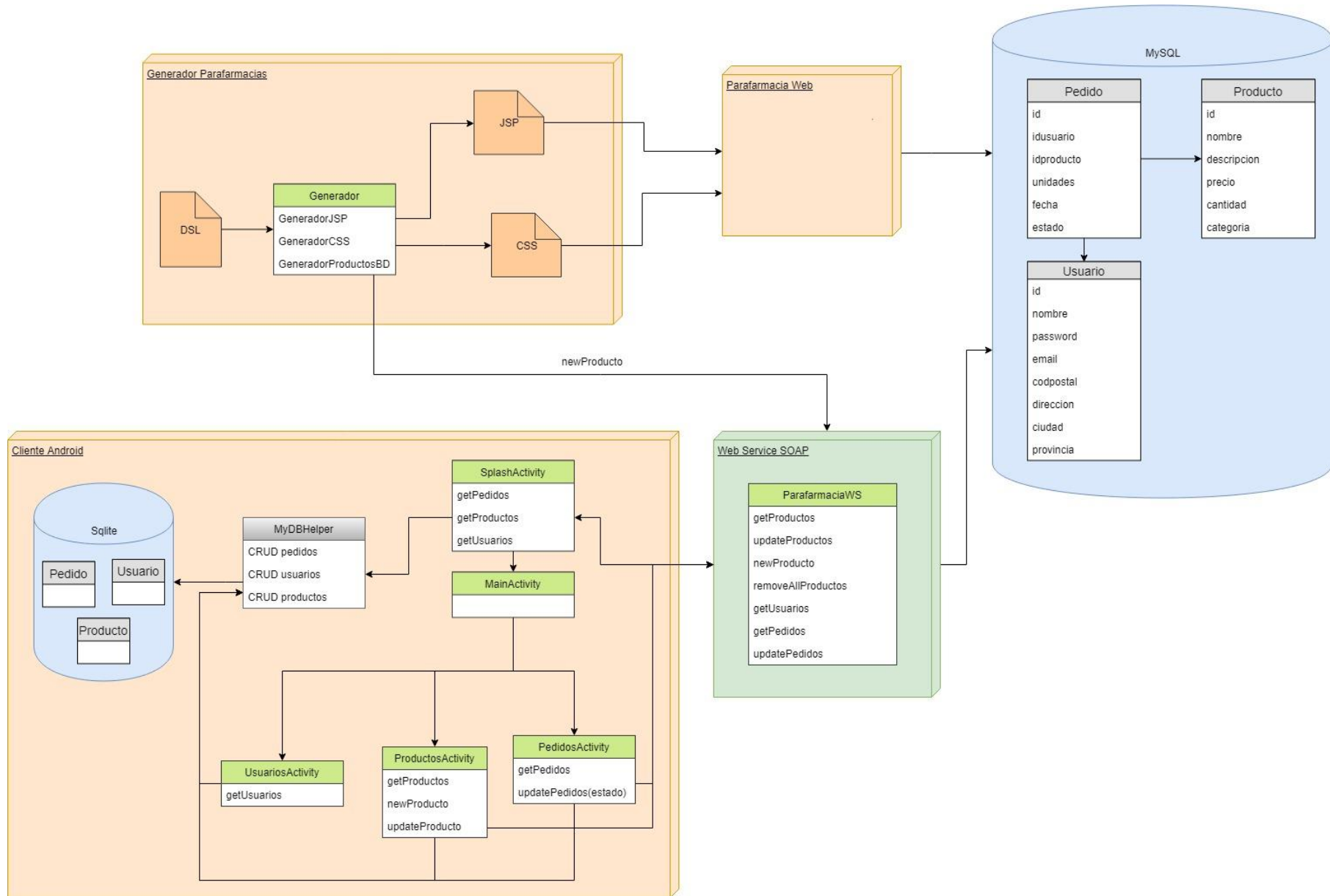


Ilustración 24 Arquitectura genera

6.1.1 Diagramas de Paquetes

6.1.1.1 Parafarmacia Web

La siguiente imagen muestra el diagrama de paquetes y clases que compone la parafarmacia web. A continuación, se pasará a describir cada uno de los paquetes.

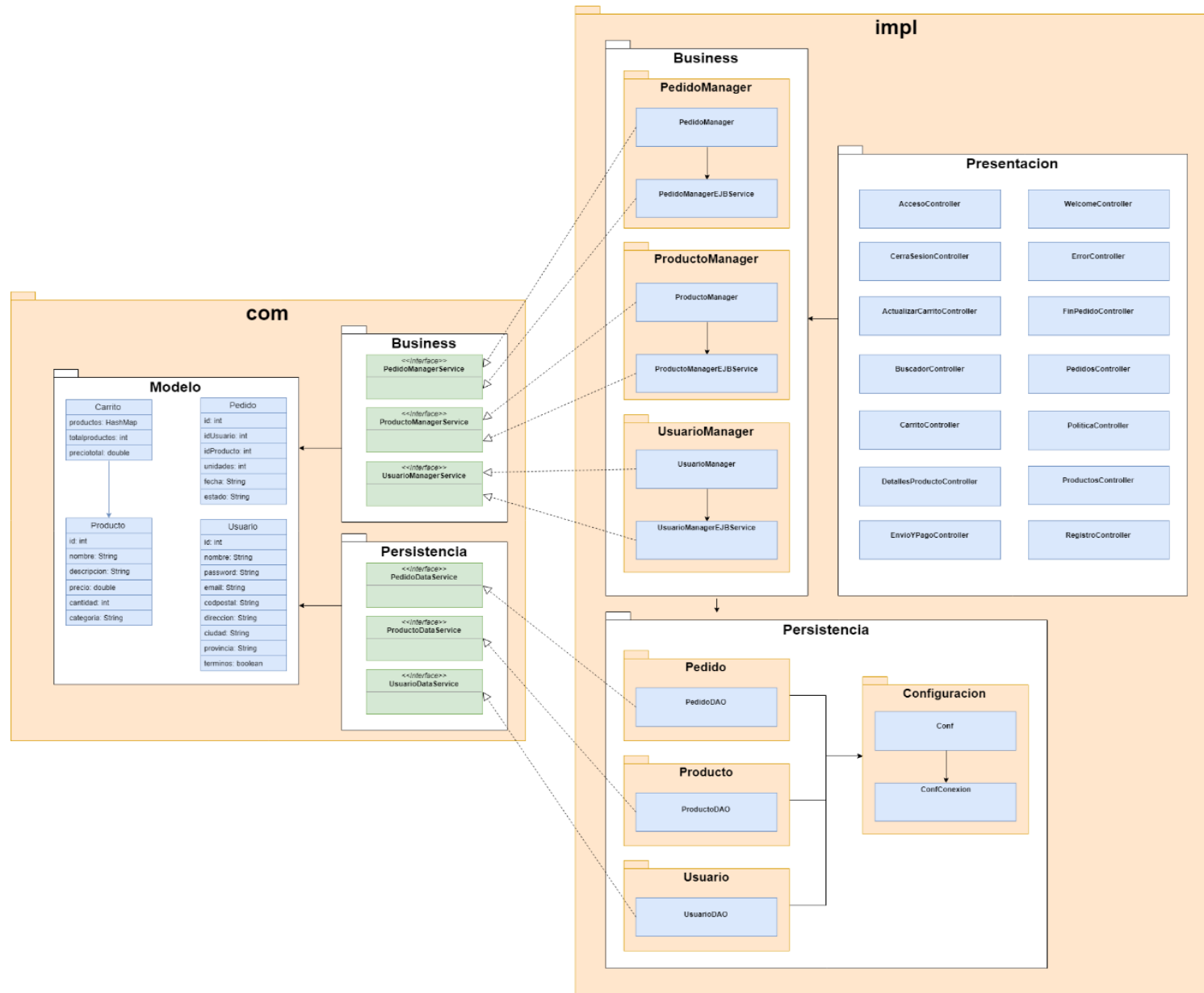


Ilustración 25 Diagrama de paquetes

6.1.1.1.1 COM

Este es un paquete general, que engloba a otro conjunto de paquetes encargados de servir las interfaces que permitirán la comunicación entre las vistas, los controladores y el modelo.

Dentro de él se pueden ver otros tres paquetes, modelo, business y persistencia.

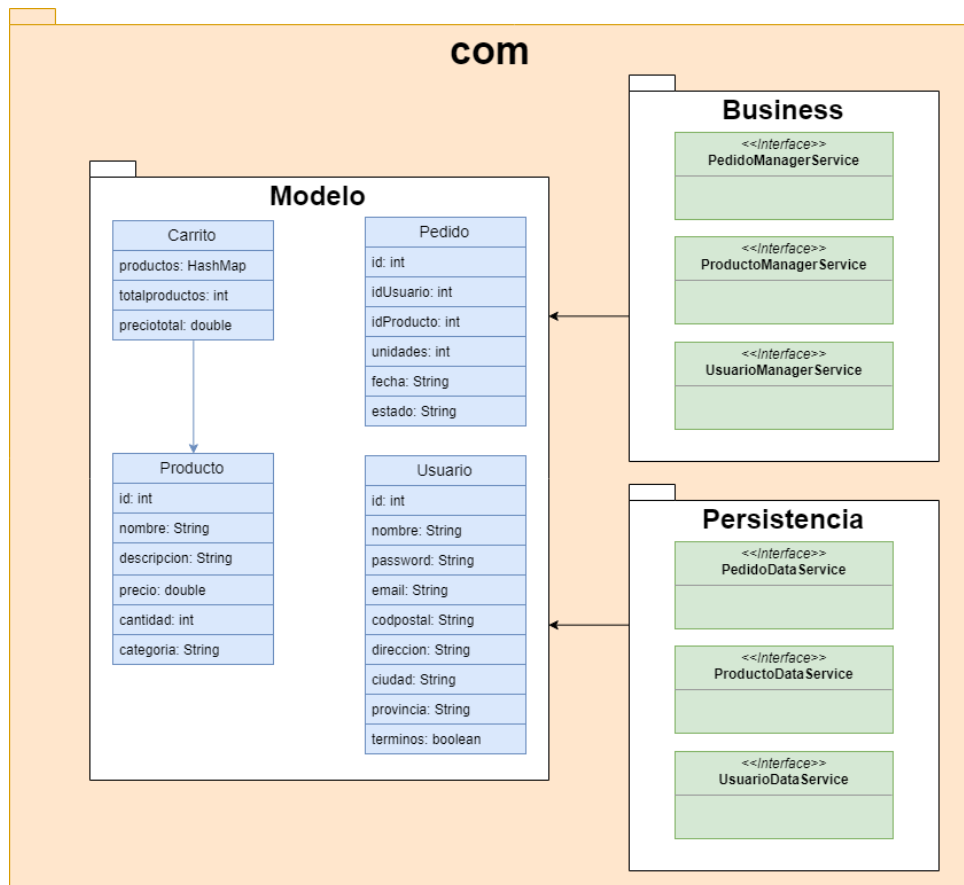


Ilustración 26 Diagrama de paquetes com

6.1.1.1.1.1 Modelo

Este paquete contiene las clases que forman el modelo con la estructura de la base de datos, usuarios, productos y pedidos además de otras entidades que serán utilizadas por otras partes del sistema, como la entidad carrito que contiene información acerca de los productos que quiere comprar el usuario y no se utilizará por la base de datos.

6.1.1.1.1.2 Business

Este paquete contiene un conjunto de interfaces que permiten al controlador comunicarse con la base de datos para obtener o guardar los datos necesarios. Se puede ver que se ha creado una interfaz para cada una de las entidades que forman el modelo.

6.1.1.1.1.3 Persistencia

Este paquete contiene un conjunto de interfaces que sirven como enlace a las clases encargadas de conectarse directamente a la base de datos.

6.1.1.1.2 Impl

Este paquete contiene un conjunto de paquetes con la implementación de las clases que forman el modelo y el controlador.

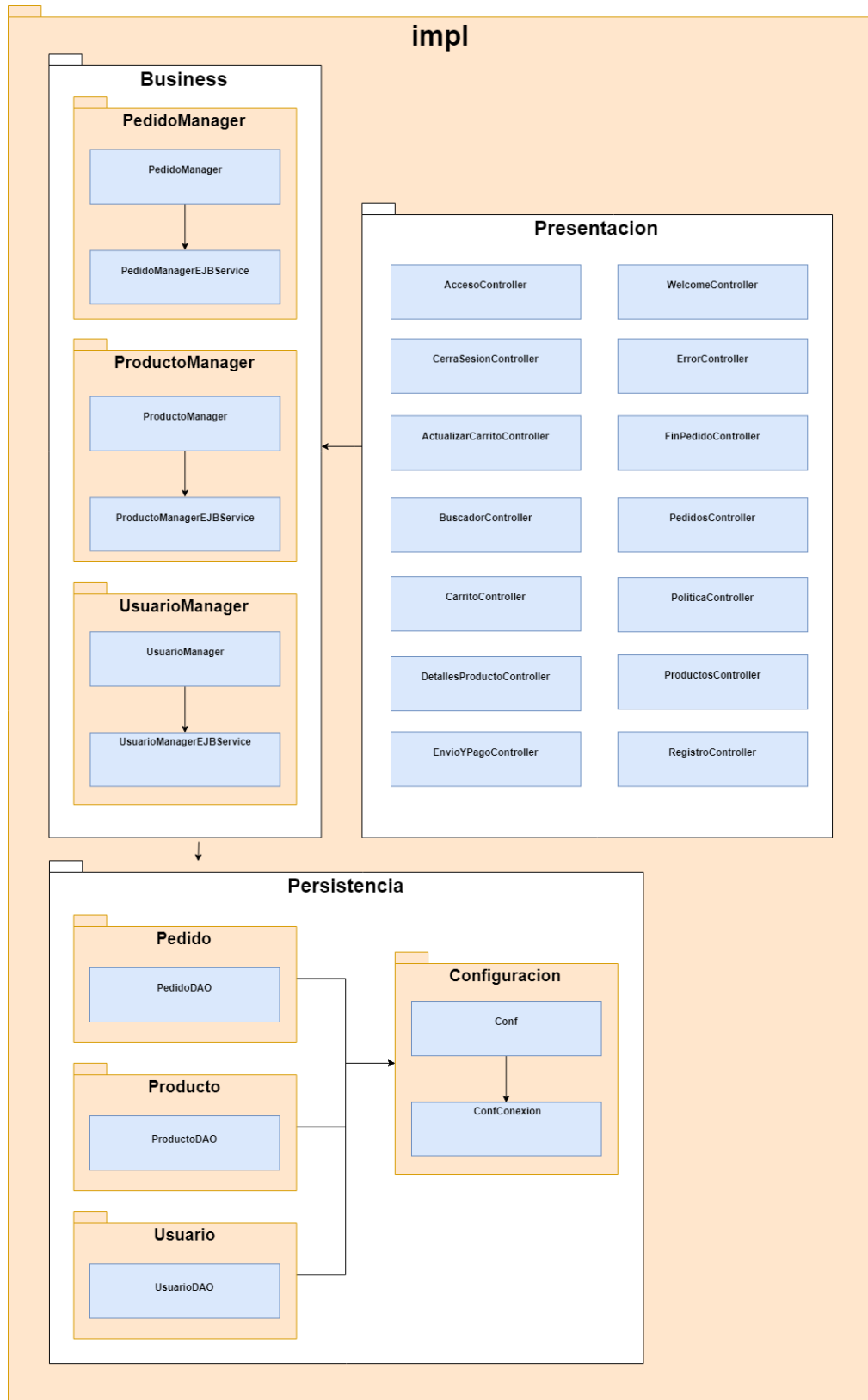


Ilustración 27 Diagrama de paquetes impl

6.1.1.1.2.1 Persistencia

Este paquete contiene la implementación de las clases que se conectan con la base de datos para obtener, crear, borrar o actualizar los pedidos, productos y usuarios. Dentro de este paquete hay otros subpaquetes que contienen cada una de las clases descritas, además de un paquete de configuración que tiene las clases encargadas de obtener información de la base de datos para que el resto se puedan conectar.

6.1.1.1.2.2 Business

Este paquete contiene las clases que serán instanciadas por los controladores para hacer uso del modelo y el acceso a los datos.

6.1.1.1.2.3 Presentación

Este paquete contiene todos los controladores de la aplicación web, encargados de acceder a los datos y pasárselos a las vistas para que el usuario final pueda interactuar con la aplicación.

6.1.1.2 Cliente Android

El siguiente diagrama de paquetes y clases contiene todos los paquetes y subpaquetes que forman la estructura del cliente Android. Se puede ver que están divididos en cuatro partes, el paquete de modelo, el de persistencia, los Activities y los adapter.

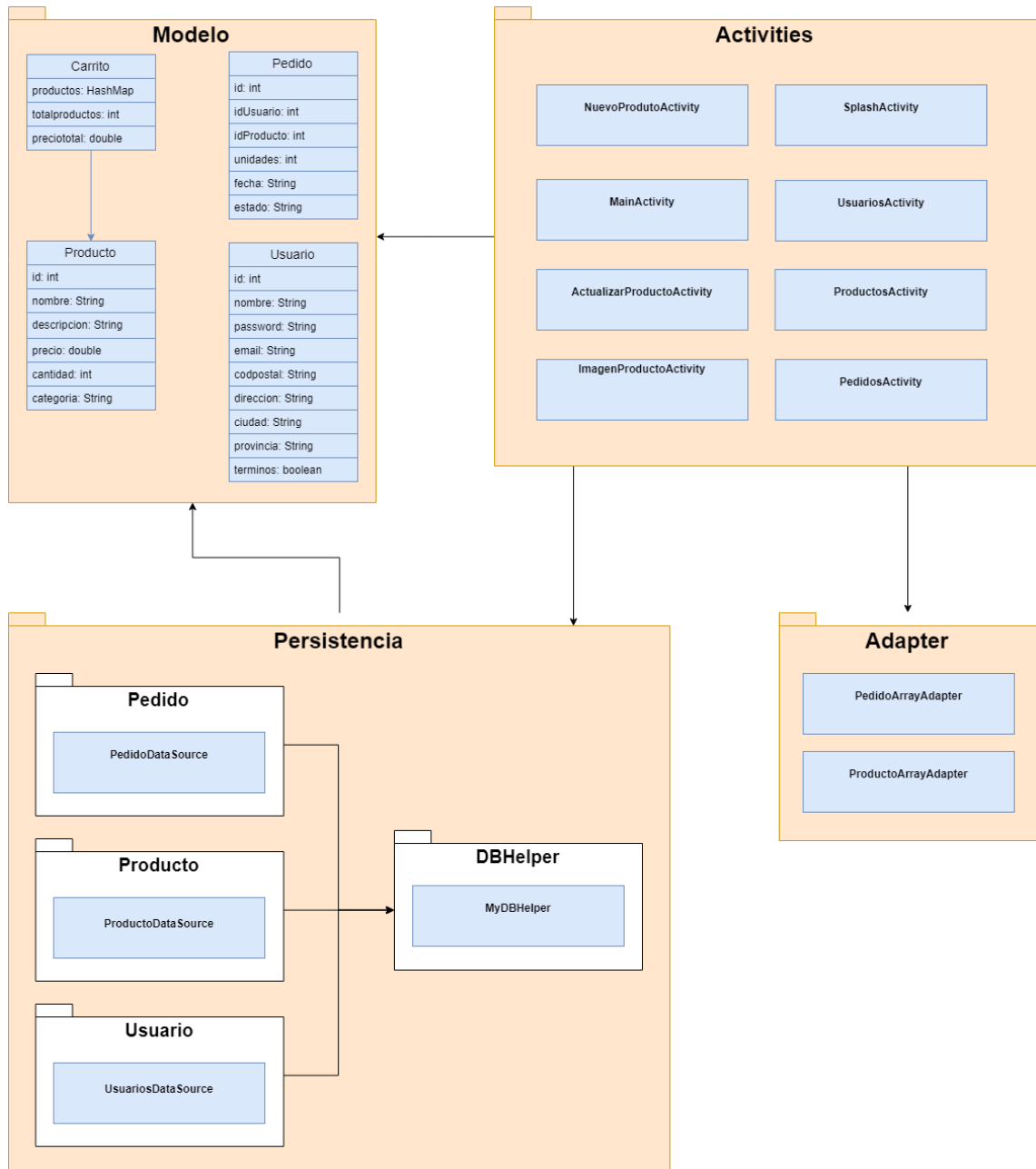


Ilustración 28 Diagrama de paquetes Cliente Android

6.1.1.2.1 Modelo

Este paquete contiene las clases que forman la estructura de datos que utilizarán las clases encargadas de acceder a la base de datos. Vemos que está formado por los usuarios, productos y pedidos, y está conectada con el paquete de persistencia, y los Activities.

6.1.1.2.2 Persistencia

Este paquete contiene varios subpaquetes, el primero, `DBHelper`, es el encargado de crear la base de datos SQLite con sus tablas correspondientes. Los otros tres subpaquetes contienen las clases que se conectan con cada una de las tablas de la base de datos y obtienen, añaden, borran o actualizan sus datos.

6.1.1.2.3 Activities

Este paquete contiene todas las clases encargadas de gestionar las vistas del cliente Android. Funciona a modo de controlador, inicializa la vista y le asigna los datos necesarios que se mostrarán al usuario.

6.1.1.2.4 Adapter

Este paquete contiene las clases que permiten crear elementos compuestos a partir de otros simples. En este caso se utilizan para visualizar en una lista los productos de manera personalizada, con la imagen, nombre, precio y categoría.

6.1.1.3 Generador

El siguiente diagrama contiene la estructura de paquetes y clases del generador. Se pueden ver tres paquetes generales, el de `servicios Web`, el `generador` y el `DSL`.

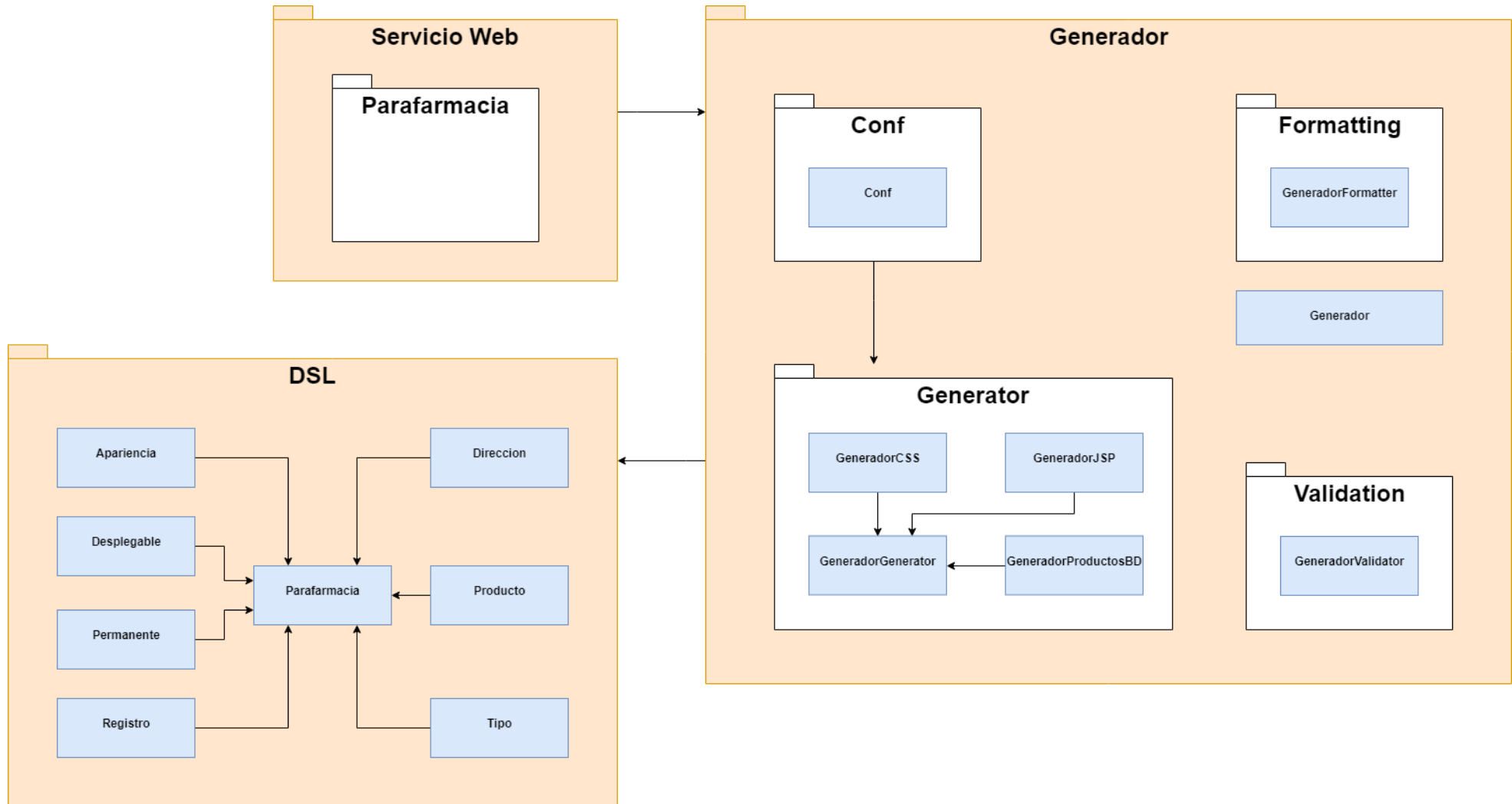


Ilustración 29 Diagrama de paquetes Generador

6.1.1.3.1 Generador

Este paquete contiene la parte más importante, en primer lugar, está la clase generador que es la encargada de crear el modelo junto a las clases que lo forman. Los paquetes *formatting* y *validation* permiten hacer validaciones en el dominio para impedir el uso de valores incorrectos y formatear el lenguaje de dominio creado. Por último, se encuentra el paquete *generator*, que contiene las clases encargadas de recibir el lenguaje de generación y generar las diferentes partes de la parafarmacia, el CSS, JSP y los productos, y el paquete *conf* que permite a estas clases hacer uso de archivos de propiedades.

6.1.1.3.2 DSL

Este paquete contiene las clases que forman el modelo del lenguaje creado por la clase generador del paquete anterior. Estas clases se crean automáticamente en función del lenguaje que queramos crear.

6.1.1.3.3 Servicio Web

Este paquete contiene el cliente del servicio web que se conecta con la base de datos de la parafarmacia web, se utiliza para insertar y borrar productos de ella.

6.1.2 Diagramas de Despliegue

El siguiente diagrama de despliegue muestra una visión general de todas las partes que componen el sistema. La parte superior comienza por el cliente, que será el usuario final de nuestra parafarmacia web. Esta parafarmacia está desplegada en un servidor, para que sea accesible por cualquier usuario. Junto a la parafarmacia web, en el servidor, también nos encontramos con la base de datos general, que servirá tanto para la parafarmacia web directamente como para el cliente Android y el generador a través de los servicios web. Los servicios web también están desplegados dentro del servidor, para servir como punto de acceso y poder utilizar así los datos de la base de datos general.

Fuera del servidor nos encontramos con el cliente Android, que será utilizado por el administrador de la tienda y se desplegará en cualquier dispositivo Android. Se conectará con el servidor a través de los servicios web por internet.

Por último, el generador, será utilizado por un desarrollador principalmente, se puede desplegar directamente en el servidor para poder generar en él la parafarmacia web o también se podrá desplegar en otros equipos que utilicen el entorno de desarrollo eclipse. Si se elige la segunda opción, será necesario copiar los archivos generados a la nueva parafarmacia web que esté alojada en un servidor.

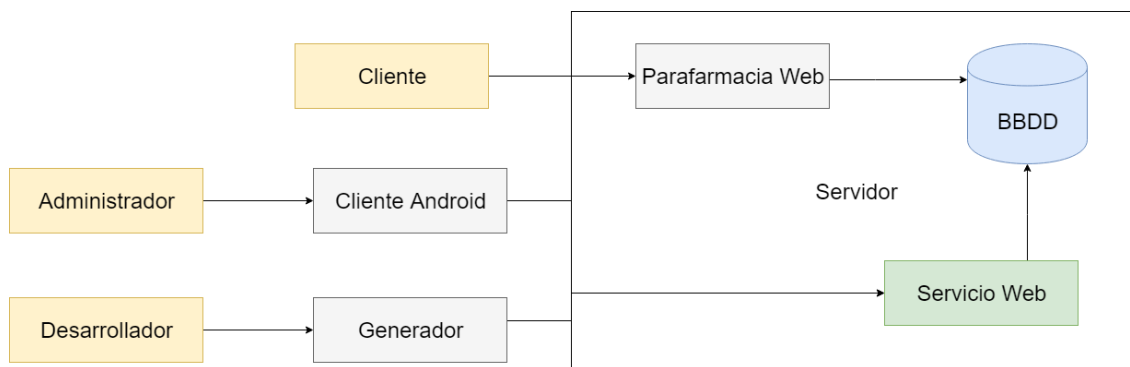


Ilustración 30 Diagrama de despliegue

6.2 Diseño de Clases

6.2.1 Diagrama de Clases

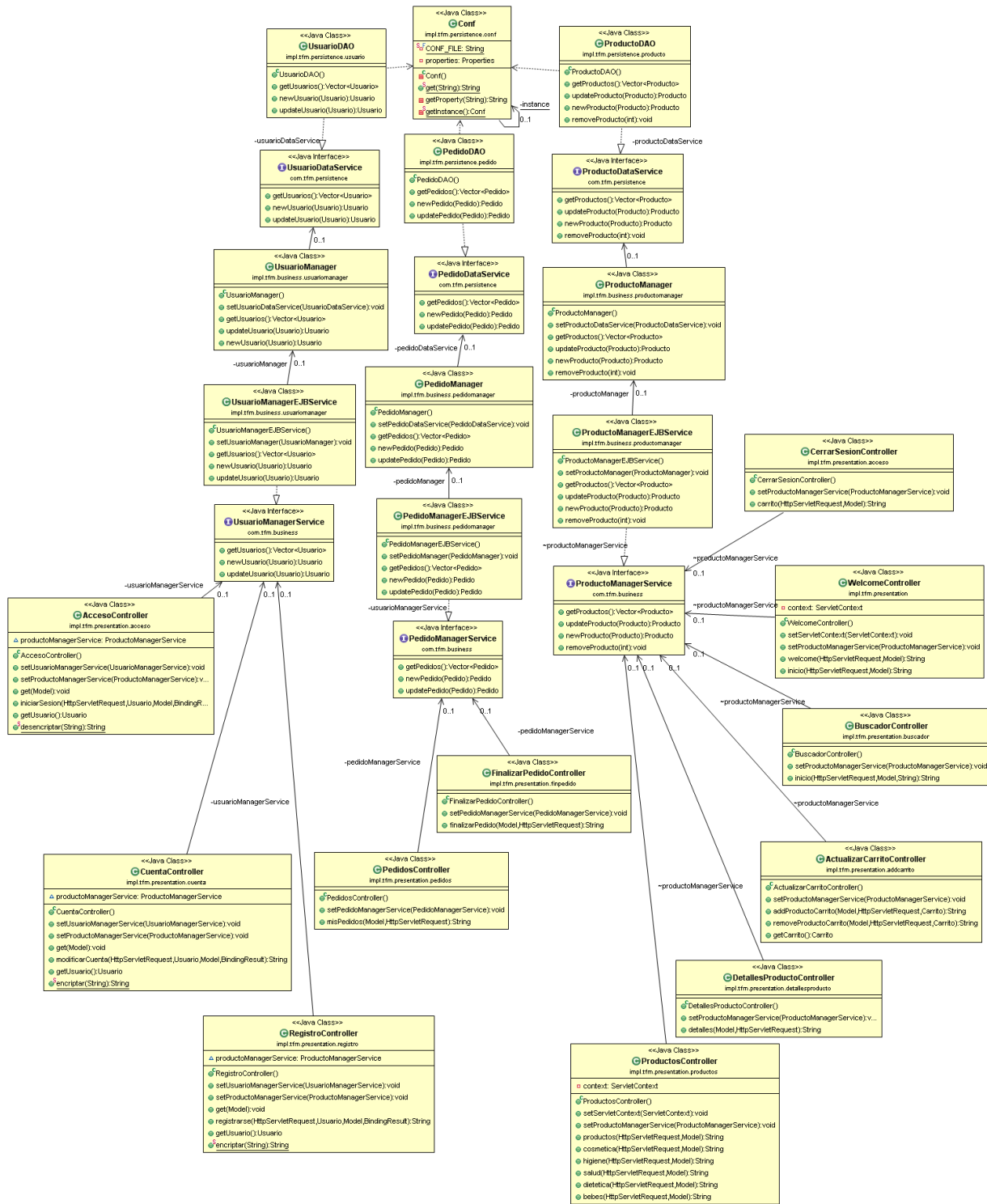
El siguiente diagrama de clases de la parafarmacia web contiene, en primer lugar, empezando por la parte inferior, las clases del modelo, pedidos, productos, usuarios y carrito. Aunque no tengan ninguna relación dibujada en el diagrama, esto se ha hecho para evitar complejidad, ya que son utilizadas por casi todas las clases, desde los controladores hasta las clases encargadas de conectarse con la base de datos para su gestión.

Encima de las clases del modelo se encuentran las clases de los controladores, que se encargan de recibir las peticiones de los usuarios y devolverles las vistas necesarias, además, se

conectan con las clases que acceden a la base de datos a través de sus interfaces para enviar a las vistas los datos solicitados.

En la última parte se encuentran las interfaces que ofrece cada una de las partes del sistema, controlador y modelo para conectarse con ellas. Se puede ver que las clases superiores son las que se conectan directamente con la base de datos utilizando la clase de configuración que contiene información de la conexión.

6.2.1.1 Parafarmacia Web



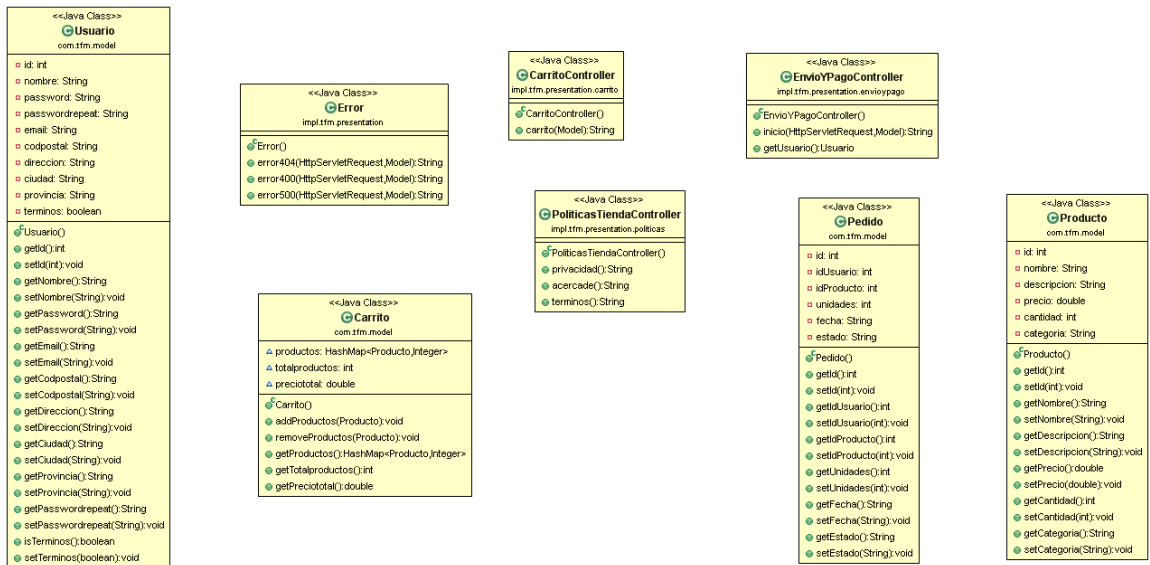


Ilustración 31 Diagrama de clases para farmacia web

6.2.1.2 Cliente Android

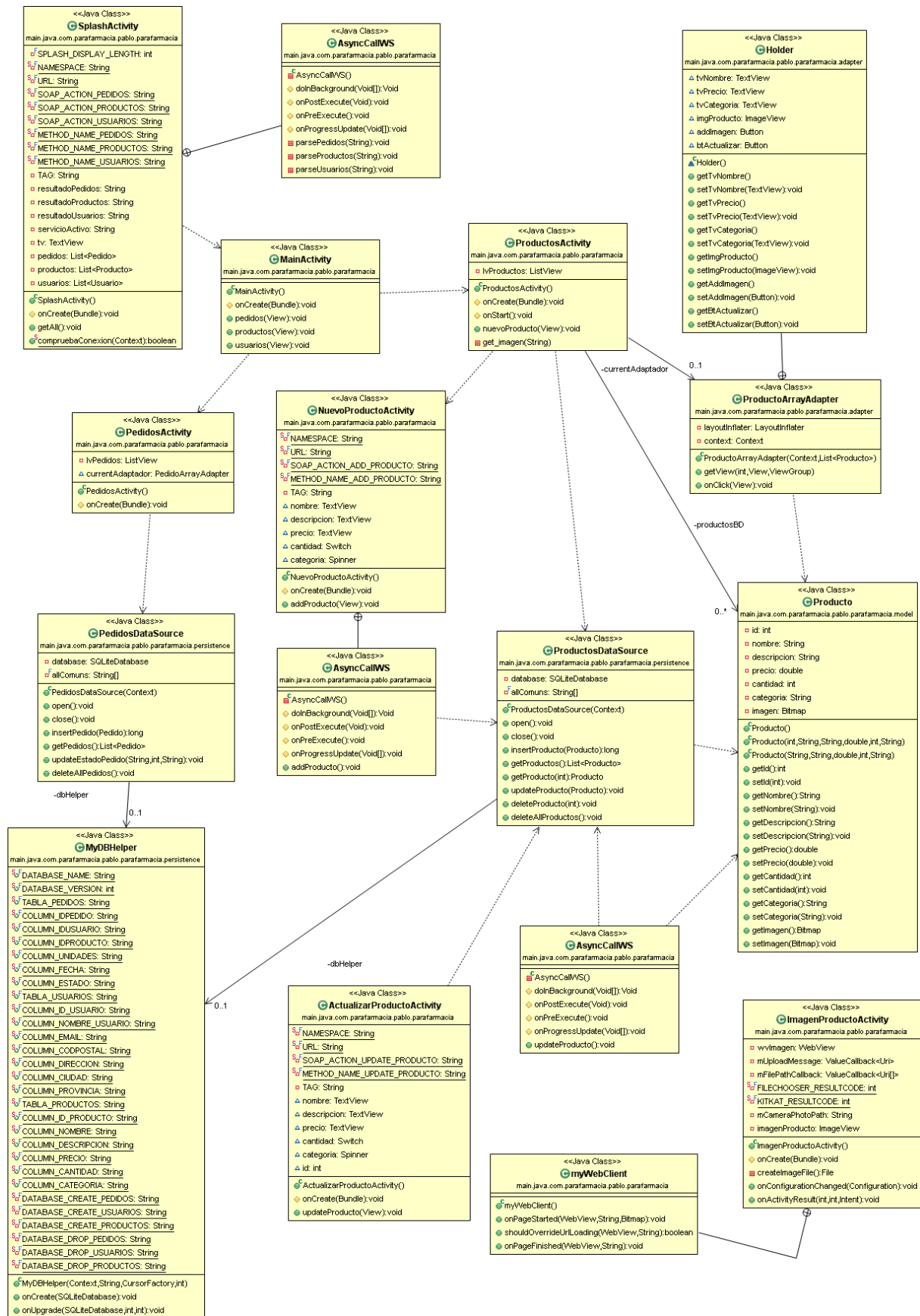


Ilustración 32 Diagrama de clases Cliente Android

El diagrama de clases del cliente Android se ha dividido en dos partes para que pueda ser visualizado mejor. La primera parte contiene las partes de usuario y pedidos y la segunda los productos. La clase `MyDBHelper` es la encargada de crear la base de datos con las tablas correspondientes, las clases encargadas de acceder a dicha base de datos son `productosDataSource`, `pedidosDataSource` y `usuariosDataSource`. Por otro lado, se encuentran las clases que conforman el modelo de datos, la clase `usuario`, clase `producto` y clase `pedido`. Por último, están las clases encargadas de gestionar las vistas, que son los `Activities`, existe uno por cada una de las vistas de la aplicación.

6.3 Diseño de la Base de Datos

La base de datos que se ha diseñado es bastante sencilla, ya que para la parafarmacia web solo es necesario almacenar información de los pedidos, los usuarios y los productos. Como se puede ver en el siguiente diagrama, los pedidos tienen relación con los usuarios y los productos debido a que necesitamos conocer quién ha realizado dicho pedido y los productos que quiere comprar.

Los datos almacenados en la base de datos del cliente Android se obtienen de la base de datos de la parafarmacia web accediendo a ellos mediante los servicios web.

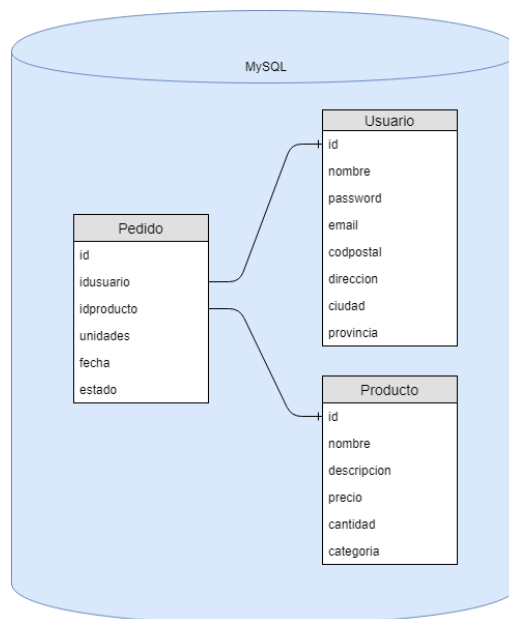


Ilustración 33 Diagrama entidad relación de la base de datos

6.4 Diseño de la Interfaz

6.4.1 Parafarmacia Web

A continuación, se pueden ver imágenes de las diferentes pantallas de la parafarmacia Web, las partes importantes ya se han explicado en apartados anteriores en los que se mostraban los sketches previos a estos diseños. Algunas de las cosas que podemos destacar aquí, son, en primer lugar, la página del menú en la que nos encontremos permanecerá con una clase activa que sirve para guiar al usuario y permitirle saber el lugar de la página en el que se encuentra. Otro detalle, al añadir nuevos productos al carrito, se mostrará un número resaltado con la cantidad de productos que tengamos en él. Los formularios se han colocado en cajas una debajo de otra de forma que se reduzca al máximo el tiempo que el usuario tarde en rellenarlo. Por último, se puede ver que la página principal es muy llamativa con muchas imágenes, intentando captar la atención del usuario para que permanezca en la página y realice sus compras.

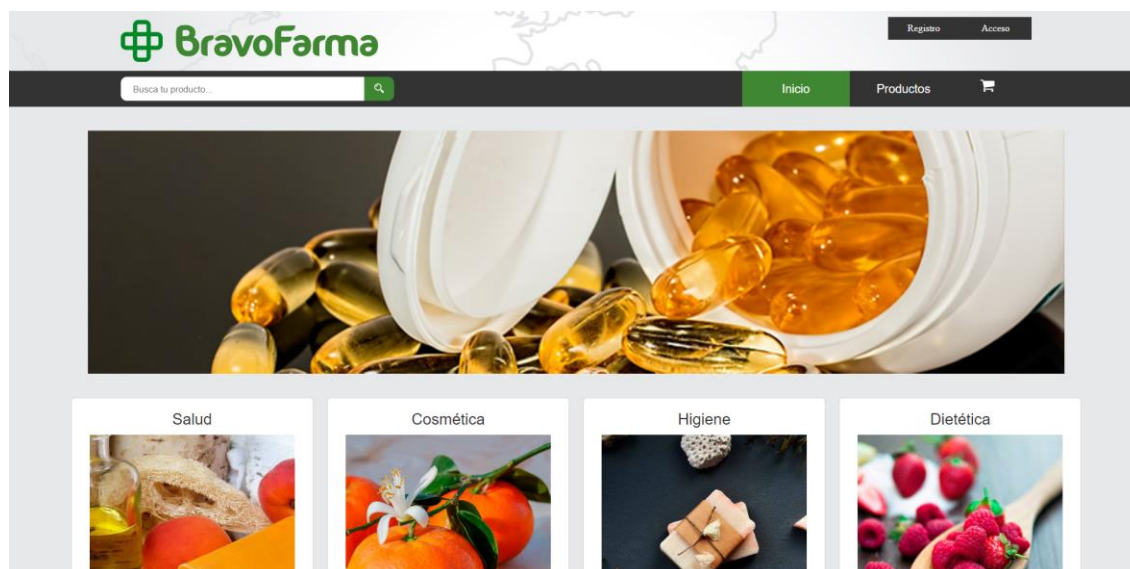


Ilustración 34 Pantalla parafarmacia web inicio

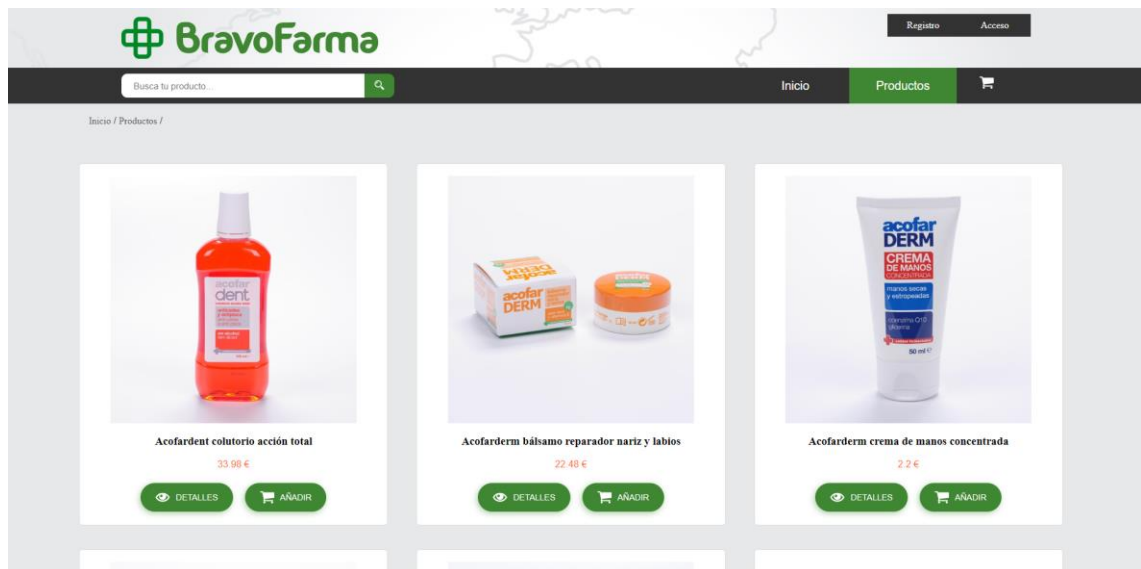


Ilustración 35 Pantalla parafarmacia web productos

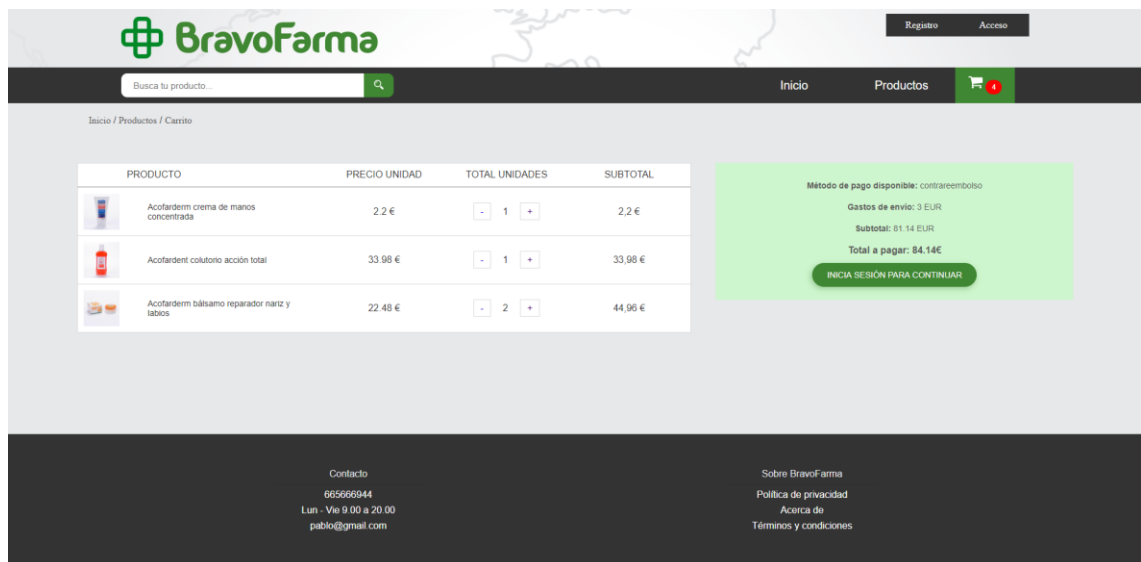


Ilustración 36 Pantalla parafarmacia web carrito

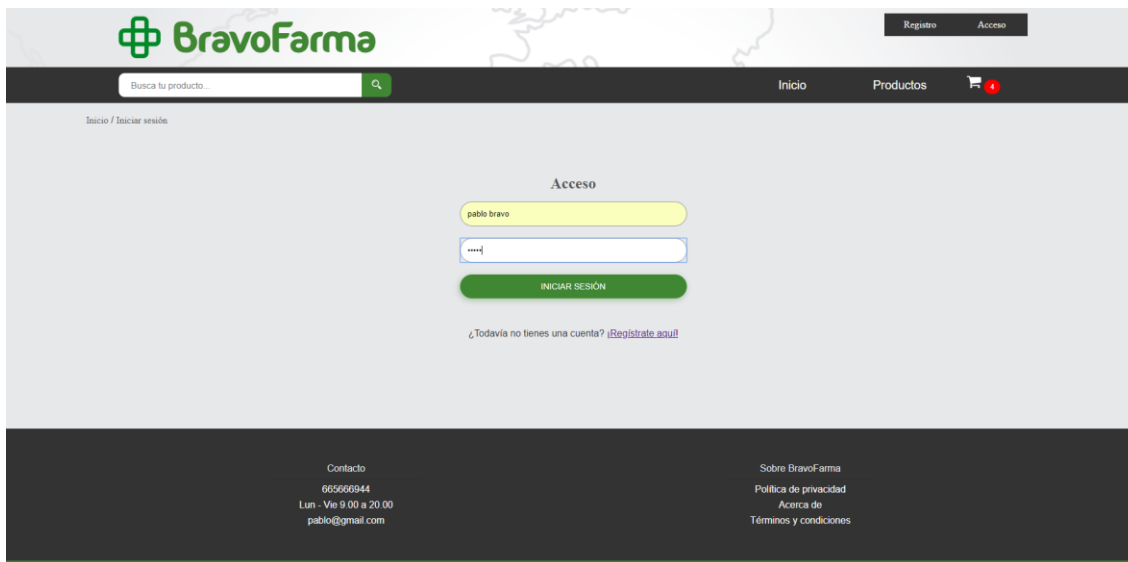


Ilustración 37 Pantalla parafarmacia web iniciar sesión

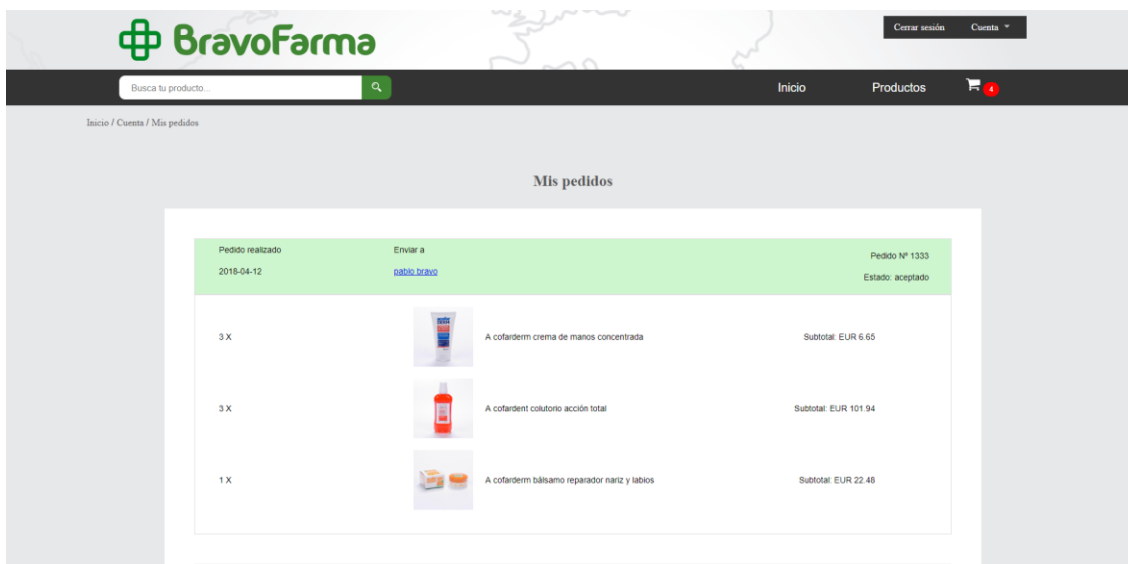


Ilustración 38 Pantalla parafarmacia web mis pedidos

BravoFarma

Cerrar sesión Cuenta

Busca tu producto...

Inicio Productos

Inicio / Cuenta /

Actualizar datos

Ilustración 39 Pantalla parafarmacia web actualizar datos

BravoFarma

Cerrar sesión Cuenta

Busca tu producto...

Inicio Productos

Inicio / Productos / Carrito / Envío y pago

Resumen de la compra

Productos			
	A cofarderm crema de manos concentrada	Unidades	2,2 €
	A cofardent colutorio acción total	Unidades	33,96 €
	A cofarderm bálsamo reparador nariz y labios	Unidades	44,96 €

Total a pagar: 84.14€ (Gastos de envío incluidos)

Una vez finalizada la compra el pedido quedará realizado y se procederá a empaquetar y enviar a la dirección solicitada por el usuario.

Datos de envío

Código postal: 33010
Dirección: calle la estación Nº 7 piso 3
Ciudad: oviedo
Provincia: asturias

Ilustración 40 Pantalla parafarmacia web envío y pago

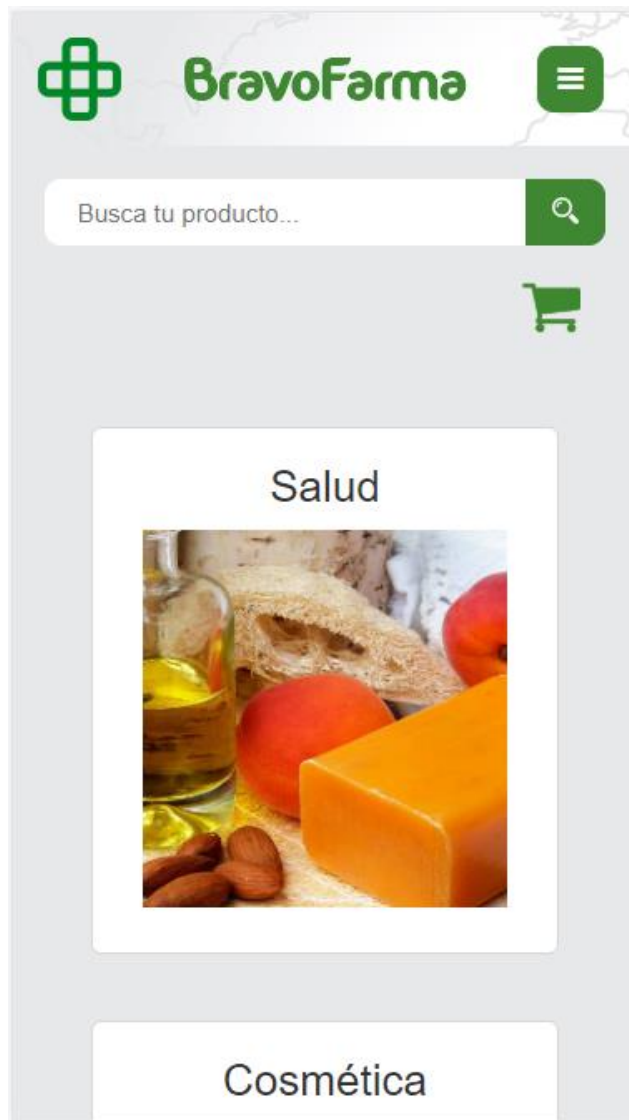


Ilustración 41 Pantalla parafarmacia web inicio responsive

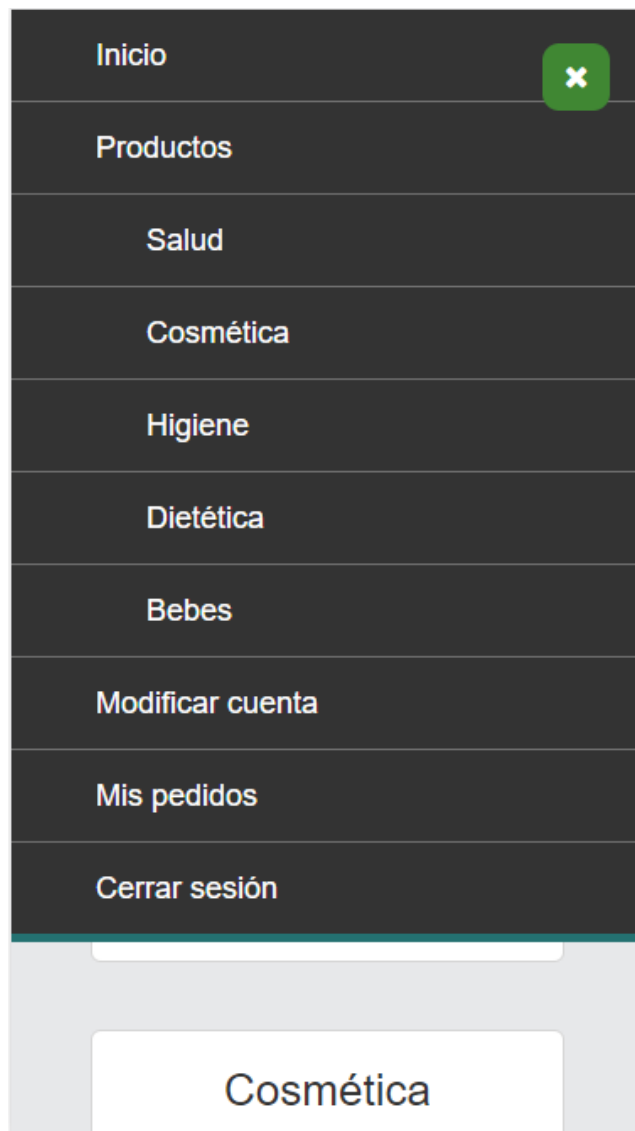


Ilustración 42 Pantalla parafarmacia web menú responsive

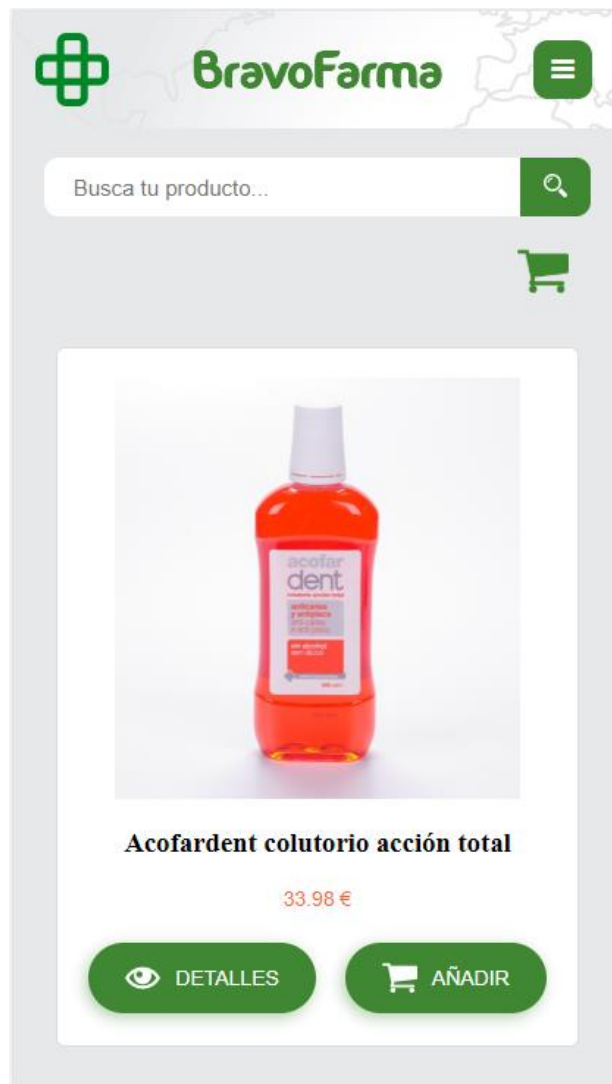


Ilustración 43 Pantalla parafarmacia web productos responsive

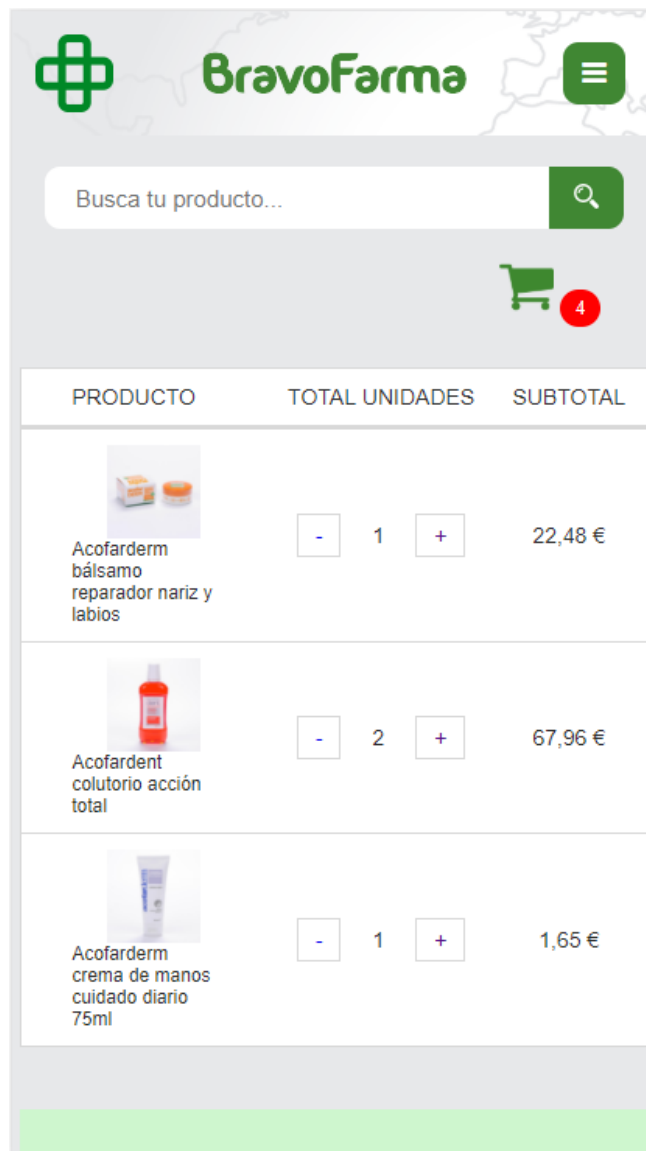


Ilustración 44 Pantalla parafarmacia web carrito responsive

6.4.2 Cliente Android

Para el diseño de estas pantallas se ha seguido todo lo descrito en uno de los apartados anteriores en los que se creaban varios sketches. Se puede ver la pantalla inicial o splash, en el que se muestra el logo y nombre, la pantalla principal con el uso de diferentes colores para guiar al usuario. Los tamaños de las letras y las alturas de los botones se han puesto siguiendo la guía indicada en el apartado de análisis, así como las flechas de navegación situadas en la parte superior izquierda.

Para el apartado de los pedidos, podemos ver que se han colocado tres botones con los diferentes estados de los pedidos. Estos botones tienen cada uno su propio color de fondo, este color no es aleatorio, si no que se ha encontrado en la página oficial de Prestashop (2) en la cual asocian a cada estado de un pedido un color identificativo.



Ilustración 45 Pantalla cliente Android splash

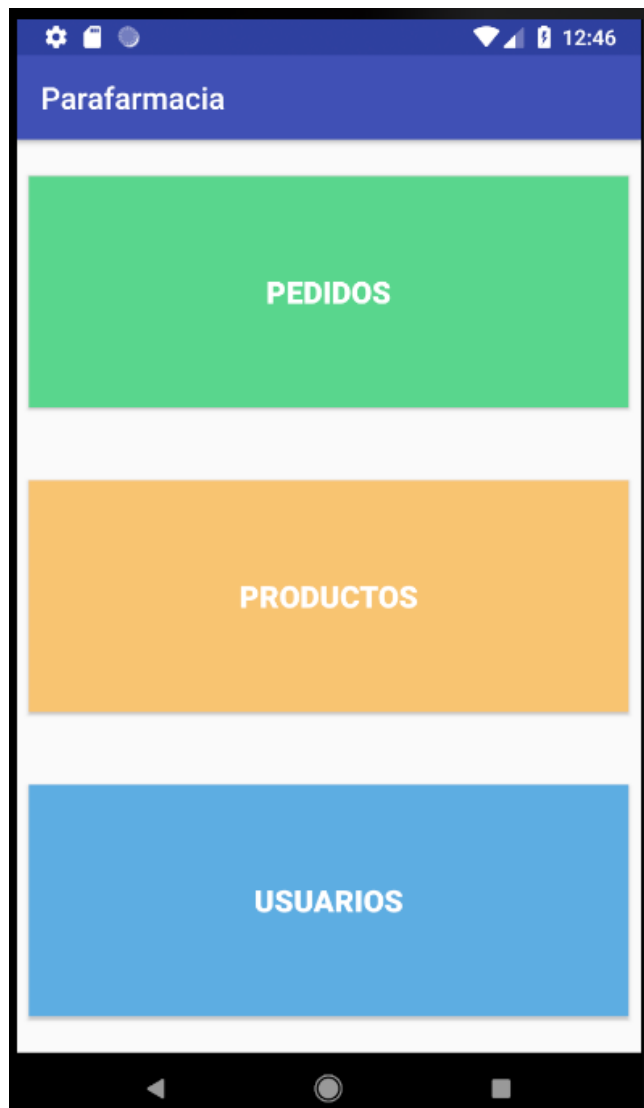


Ilustración 46 Pantalla cliente Android principal

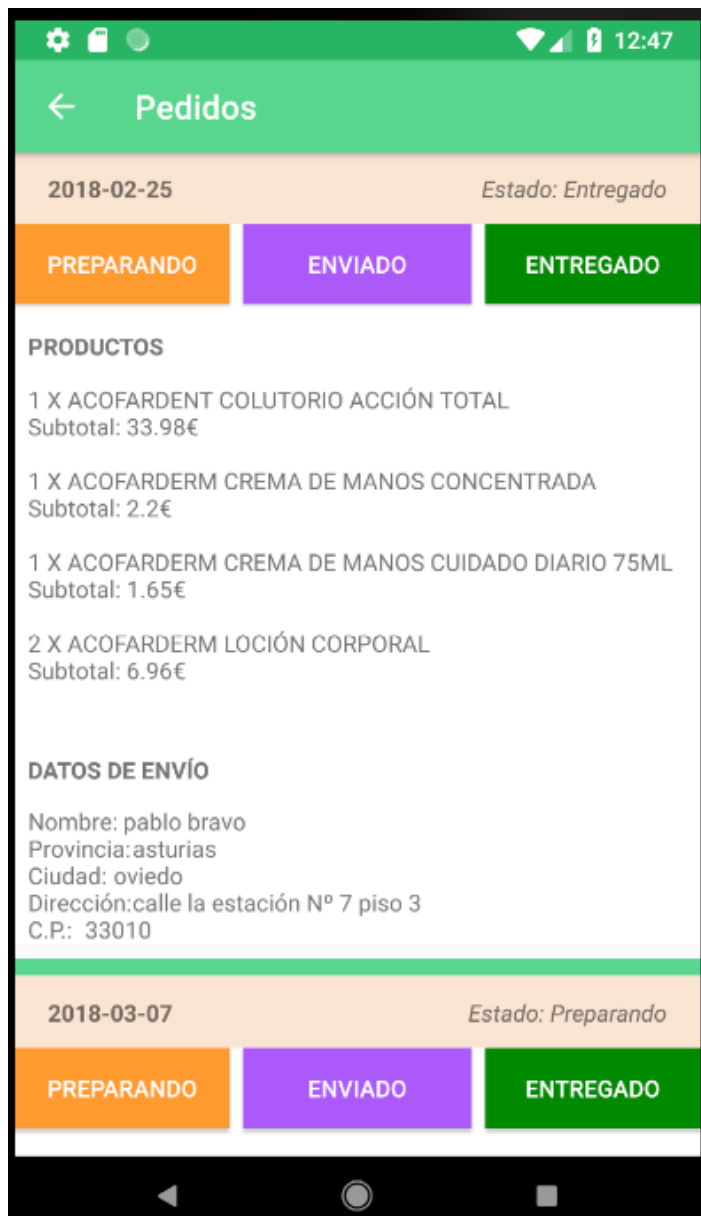


Ilustración 47 Pantalla cliente Android pedidos



Ilustración 48 Pantalla cliente Android productos



Ilustración 49 Pantalla cliente Android usuarios

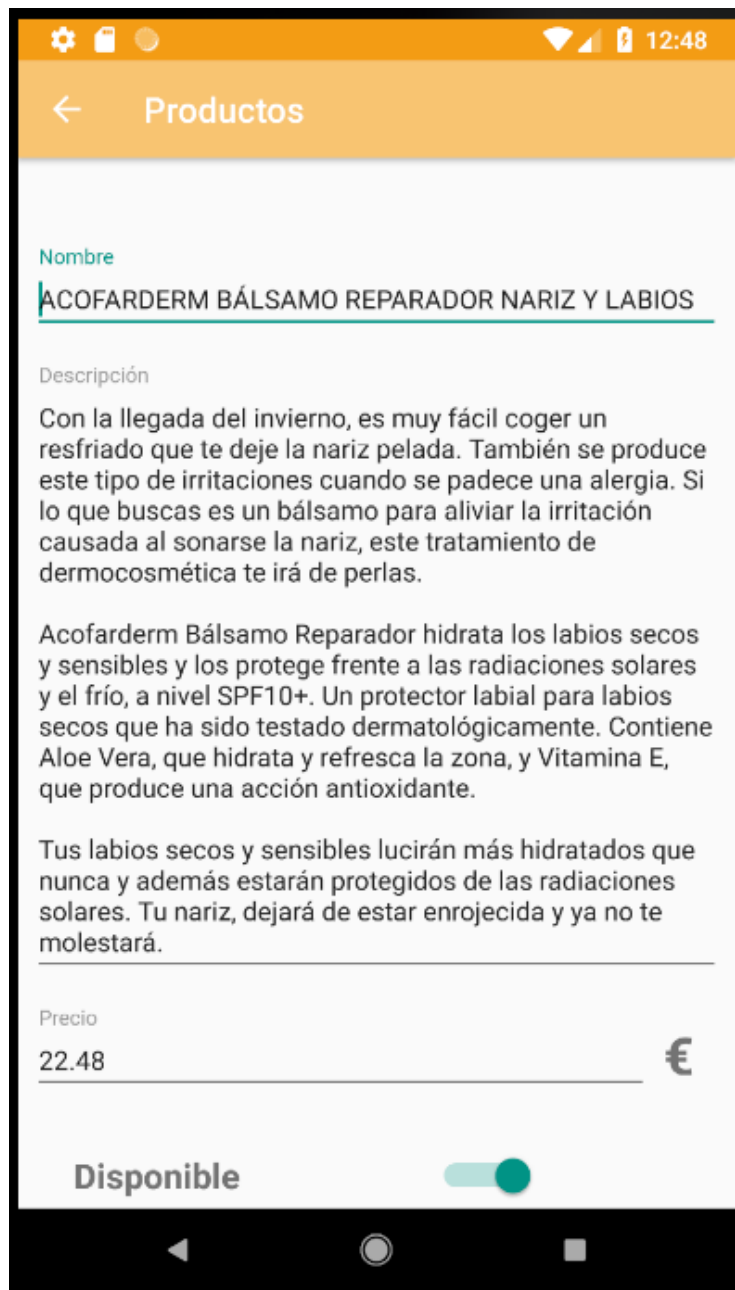


Ilustración 50 Pantalla cliente Android actualizar producto

6.4.3 Generador

Para desarrollar el generador se ha diseñado un lenguaje propio que tiene una serie de características que se explican a continuación.

Cada parafarmacia que se quiera generar debe tener su identificador propio, “Parafarmacia”, seguido de unas llaves entre las cuales se escribirá todo el código de personalización. La parafarmacia tiene varios aspectos generales que se pueden personalizar, el nombre de la parafarmacia, el titular, el tipo de menú, la apariencia, las categorías de los productos, los productos, los métodos de pago y los datos del usuario administrador. Cada una de estas partes tiene asignada una palabra clave representativa y se definen entre símbolos de mayor y menor, incluyendo entre ellos el resto de código que permite personalizar dicha parte. Dentro de cada uno de los aspectos generales podemos definir elementos simples o elementos compuestos. Los elementos compuestos, como una dirección, formada por la calle, la ciudad y el país o un producto, formado por nombre, descripción, precio, cantidad y categoría, se encierran entre corchetes. Los elementos simples son los que el usuario puede modificar para generar la parafarmacia a su gusto, se separan entre ellos mediante dos puntos. Por último, nos encontramos los elementos simples especiales, por ejemplo, los códigos de los colores o los tamaños de las letras, que por sí solos no son representativos, por esta razón, estos elementos se declaran escribiendo delante de ellos, utilizando un punto para su separación, una palabra reservada descriptiva para cada elemento.

```

Parafarmacia
{
  < Nombre = "nombreParafarmacia" : "fuenteTitulo" >
  < Titular = "nombreTitular" :
  [
    Direccion =
      "calle" :
      "ciudad" :
      "pais"
    ] :
    "33010" :
    [Registro = 33 : 33 : 33 : 33] :
    p@p.com :
    66666666
  >
  < Menu = Permanente : #colorMenu: 33px >
  < Apariencia =
  [
    Color =
      principal.#55 :
      principalClaro.#55 :
      secundario.#55 :
      fondo.#55 :
      fondoProductos.#55 :
      precios.#55 :
      texto.#55 :
      titulo.#55
    ]
    [
      Texto =
        normal.33px :
        grande.33px :
        titulo.33px :
        menu.33px
      ]
    [
      Margen = 33%
    ]
  ]
  >
  < Categorias = Bebes >
  < Productos = productosPorFila.33
  [
    Producto =
      'nombre':
      'descripcion':
      33.33:
      33:
      Bebes
    ]
  ]
  >
  < MetodoPago = Paypal >
  < Usuario = "nombreUsuario" : p@p.com : "password" >
}

```

Ilustración 51 Pantalla lenguaje Generador

Capítulo 7. Implementación del Sistema

7.1 Estándares y Normas Seguidos

A continuación, se describen una serie de estándares y normas que afectan al proyecto y que se han tenido en cuenta a la hora de desarrollarlo.

7.1.1 W3C

W3C (3) es una organización encargada de desarrollar los estándares aplicados a la web, permite comprobar ciertos aspectos de accesibilidad para las versiones de dispositivos móviles y para las versiones de ordenador.

7.1.2 Android Design

Android Design (1) es una estándar creado por Google para los desarrolladores de Android, que permite seguir una serie de patrones que se empiezan a seguir en todas las aplicaciones. Éste estándar no es de obligatorio cumplimiento, pero si es recomendable seguirlo.

7.1.3 Estándares de programación

Para desarrollar el código Java que compone la parafarmacia web se han seguido una serie de estándares de programación (4). Alguna de las cosas que aparecen en estos estándares son la organización de ficheros, comentarios de inicio en los ficheros fuente, como deben colocarse los paquetes del proyecto, declaraciones de clases e interfaces, sangrías a respetar, longitudes máximas de línea o respetar los espacios en blanco.

7.1.4 Ley Orgánica de Protección de Datos de Carácter Personal

La Ley Orgánica de Protección de Datos de Carácter Personal (5) se aplica en este proyecto debido a la necesidad de registrar datos sensibles de usuarios en la web. Estos datos serán recogidos con la finalidad de permitir al usuario realizar compras en la parafarmacia. Por esta razón, es necesario cumplir con los artículos que dispone la ley.

7.1.5 Ley de Servicios de la Sociedad de Información

La Ley de Servicios de la Sociedad de Información (6) engloba, además de la contratación de bienes y servicios por vía electrónica, el suministro de información mediante dicho medio, transmisión de datos por redes de telecomunicación, el alojamiento en servidores de información, provisión de instrumentos de búsqueda o enlaces a otros sitios que aparezcan en Internet siempre que representen una actividad económica para el prestador de los servicios. Por esta razón, la parafarmacia web debe cumplir con esta ley ya que representa una actividad económica.

7.2 Lenguajes de Programación

La siguiente lista contiene los lenguajes de programación utilizados para desarrollar el sistema junto a la especificación del subsistema en el cual se ha utilizado.

7.2.1 JAVA

Java es un lenguaje de programación de propósito general orientado a objetos, su intención es poder reutilizar el código para que se pueda ejecutar en cualquier dispositivo sin necesidad de volver a compilarlo.

Este lenguaje fue creado por James Gosling de Sun Microsystems, que actualmente es conocida como Oracle, y se lanzó al mercado en el año 1995 como componente fundamental de la plataforma Java de Sun Microsystems.

En este proyecto se utiliza el lenguaje en diferentes plataformas:

- **Eclipse y Xtext:** la plataforma Eclipse trabaja fundamentalmente con el lenguaje Java como ya se ha comentado anteriormente. El generador de parafarmacias utiliza esta plataforma con este lenguaje para su desarrollo.
- **Android Studio:** la plataforma de desarrollo de aplicaciones Android también utiliza Java como lenguaje principal, por lo que el cliente Android se ha realizado utilizando este lenguaje.
- **Spring MVC:** el framework utilizado para desarrollar la parafarmacia web, que utiliza Java para su implementación.

7.2.2 HTML 5

HTML, o *Hypertext Markup Language* como se conoce en inglés, es el lenguaje de marcado para el desarrollo de páginas web. Es un estándar que sirve de referencia para el software, ya que define la estructura básica que va a tener la web.

Este lenguaje está a cargo del *World Wide Web Consortium*, también conocida como **W3C** o consorcio *WWW*, organización dedicada a la estandarización de diferentes tecnologías utilizadas en el desarrollo web.

En este proyecto se ha utilizado la última versión que disponemos actualmente, la 5.

```
<!doctype html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1" />
<meta name="MobileOptimized" content="width" />
<meta name="HandheldFriendly" content="true" />
<meta content="Pablo Bravo Mediavilla" name="author" />
<link rel="shortcut icon"
      href="<c:url value="/resources/img/logo.ico"/>" />
<title>Acceso</title>
<link href="<c:url value="/resources/css/style.css" />" rel="stylesheet">
</head>
<body>
  <div class="wrapper">

    <jsp:include page="generic/menu.jsp" />
    <div class="main">
      <div class="subcabecera">
        <p>
          <a href='inicio'>Inicio</a> /
          <a href='acceso'>Iniciar sesión</a>
        </p>
      </div>
      <div class="formacceso">
        <h2>Acceso</h2>
        <p>
          <c:out value="\${error}"></c:out>
        </p>
        <form:form commandName="usuario">
```

Ilustración 52 Ejemplo de código HTML utilizado en la parafarmacia web junto a JSP

HTML se ha utilizado para el desarrollo de la parte visual de la parafarmacia web, incluyéndose dentro de los archivos propios de un proyecto web para Java, los *JavaServer Pages (JSP)*.

7.2.3 CSS 3

CSS o, como también se le conoce, *hoja de estilo en cascada*, es un lenguaje utilizado para definir y crear la presentación de un documento que se ha escrito con HTML o XML.

El World Wide Web o W3C mencionado en el apartado del lenguaje HTML, también es el encargado de realizar la especificación de las hojas de estilo, que servirán de estándares para los navegadores que carguen la página web.

Este lenguaje se creó para tratar de separar la parte visual de una web, de la estructura.

```
/* General */
body {
    margin: 0 auto;
    padding: 0;
    width: 100%;
    background-color: var(--background-color);
}

h1 {
    margin: 0 auto;
    padding: 0;
}

hr {
    height: 1px;
    background-color: var(--color-text);
    width: 100%;
    border: none;
}

p {
    margin: 0 auto;
    padding: 0;
    font-family: var(--familia-texto);
}
```

Ilustración 53 Ejemplo de código CSS utilizado en la parafarmacia web

En este proyecto se ha utilizado el lenguaje únicamente en una parte, la parafarmacia web. Su función es crear la parte visual y permitir colocar los elementos del HTML de forma que sean visualmente atractivos. Al ser una versión mejorada del proyecto del grado, se han incluido elementos nuevos que permiten realizar un diseño mucho más sencillo y limpio, como la definición de variables o el posicionamiento de los componentes mediante flex-box.

7.2.4 JavaScript

JavaScript es un lenguaje de programación interpretado, orientado a objetos y dinámico. Forma parte del estándar *ECMAScript*, se utiliza normalmente en aplicaciones web dinámicas del lado del cliente, para realizar mejoras en las interfaces de usuario. Aunque también existe del lado del servidor, en este proyecto se ha utilizado únicamente para realizar modificaciones en la interfaz de usuario de la parafarmacia.

7.2.5 PHP

PHP o *Hypertext Preprocessor* es un lenguaje de programación de código abierto utilizado para el desarrollo web y que se puede incrustar en las páginas HTML. En este proyecto, su función es muy breve y básica, permite añadir nuevas imágenes de los productos a la parafarmacia o actualizar las actuales. El cliente Android es el encargado de utilizar el código para gestionar los productos.

7.3 Herramientas y Programas Usados para el Desarrollo

7.3.1 Eclipse

Eclipse es una plataforma de software formada por un grupo de herramientas para programar código multiplataforma, fue desarrollada por IBM. Para este proyecto se ha utilizado la versión Oxygen, cuyo lanzamiento fue en junio de 2017.

Se ha elegido utilizar esta plataforma debido a que durante el periodo de estudio en la universidad ha sido la herramienta utilizada por defecto, por lo que no es necesario dedicar un gran esfuerzo para aprender a usarla.

Este entorno de desarrollo se ha utilizado en varias partes del proyecto, en primer lugar, para crear el generador de parafarmacias, ya que Xtext es un plugin que se encuentra disponible desde el entorno, además, los servicios web que permiten la conexión con la base de datos también se han desarrollado utilizando Eclipse.

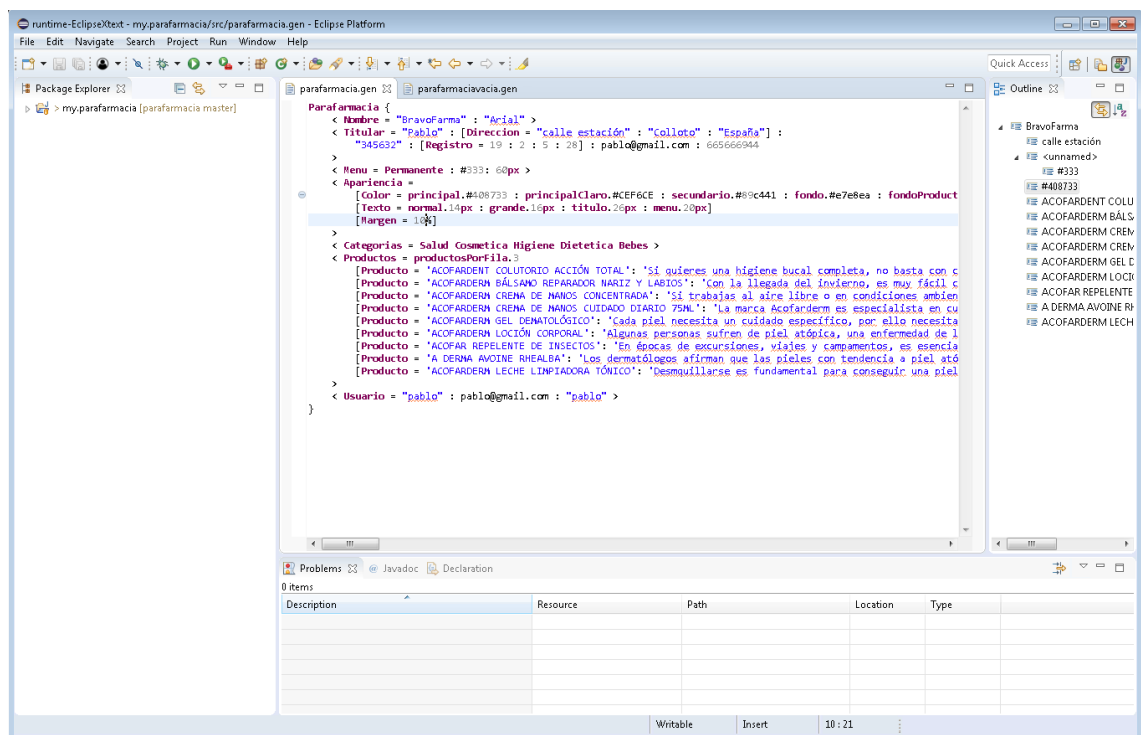


Ilustración 54 Ejemplo ventana de Eclipse con el lenguaje de generación

7.3.2 Spring Tool Suite

Spring Tool Suite, también conocido por **STS**, es un entorno que permite implementar, depurar, ejecutar y desplegar aplicaciones *Spring*, permitiendo integrar servidores, *GitHub* (explicado en esta sección) o *Maven*, para gestionar y construir el proyecto mediante sus dependencias. Es de código abierto y proporciona varias herramientas para los desarrolladores de aplicaciones webs, como la visión en tiempo real de los parámetros de rendimiento para identificar y diagnosticar problemas fácilmente.

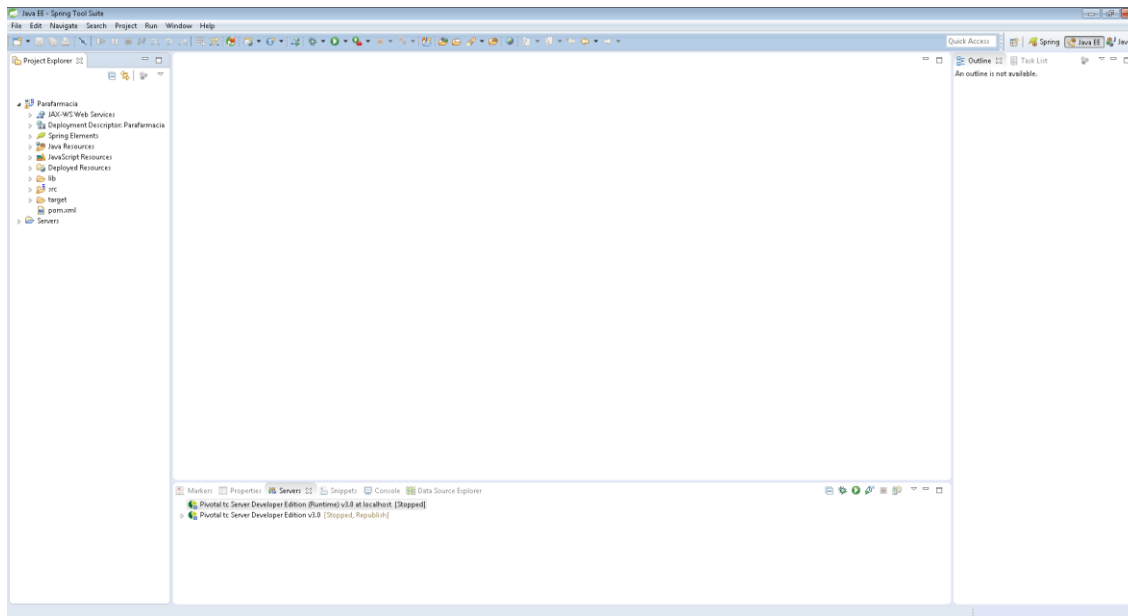


Ilustración 55 Pantalla inicial Spring Tool Suite con el proyecto parafarmacia

El uso que hemos dado a este entorno durante el proyecto ha sido para crear la parafarmacia web, ya que da soporte para construir aplicaciones utilizando el *framework Spring*, y además permite desarrollar utilizando el lenguaje de programación JAVA.

7.3.3 Android Studio

Android Studio es un entorno de desarrollo creado para la plataforma de Android, se lanzó en mayo de 2013 y a partir de ahí a remplazado a la plataforma de Eclipse en el desarrollo de aplicaciones Android, ya que se centra únicamente en el desarrollo de aplicaciones para sistemas operativos Android y se encuentra completamente equipada con todas las opciones necesarias para los desarrolladores.

Para el proyecto se ha utilizado inicialmente la versión 3.0.1, que se ha ido actualizando debido a las nuevas versiones que se han lanzado al mercado y las mejoras que se han incluido, por ejemplo, aumentar la velocidad de procesamiento, lo que permite trabajar simultáneamente con más programas, cosa que en sus versiones previas era casi imposible. Otro de los grandes avances que se han incluido en las actualizaciones ha sido la visualización de las interfaces de usuario junto con el nuevo *Constraint Layout* que permite de una forma muy cómoda distribuir los elementos.

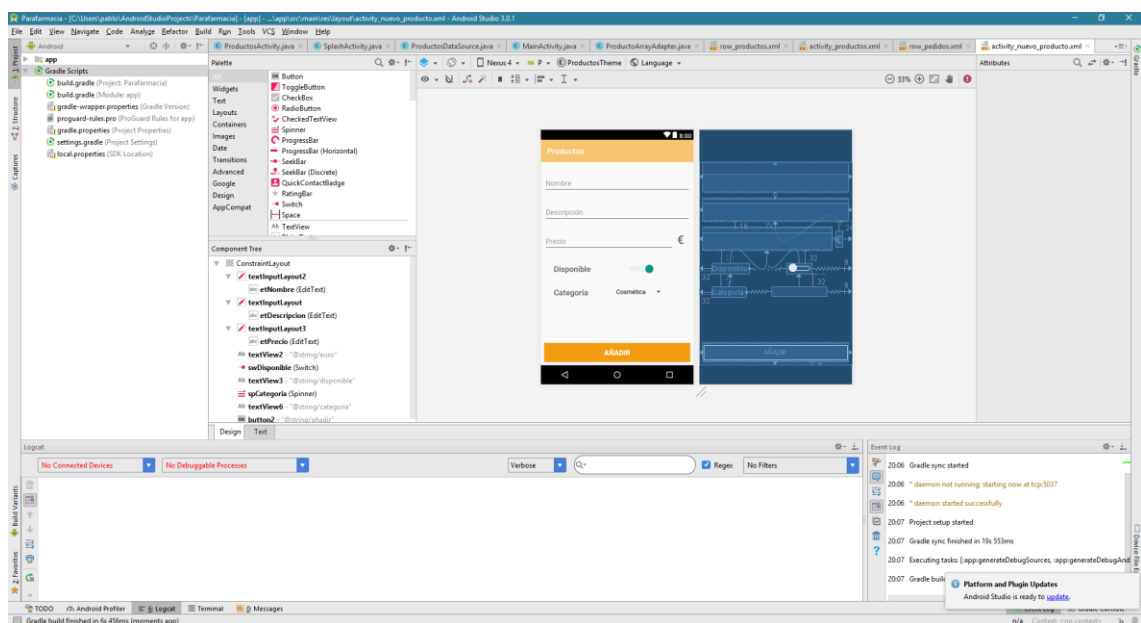


Ilustración 56 Pantalla de ejemplo de Android Studio

Su finalidad en el proyecto ha sido el desarrollo del cliente Android, para la gestión de tiendas online. Ha permitido crear el código necesario y las pantallas correspondientes para permitir a los usuarios que la utilicen ver los usuarios registrados en su tienda, los productos, así como añadirlos o eliminarlos y los pedidos.

7.3.4 Illustrator

Illustrator es un editor de gráficos vectoriales que pertenece a la compañía Adobe.

En este proyecto se ha utilizado la última versión disponible en su versión de prueba, que se conoce como CC 22.0.1.

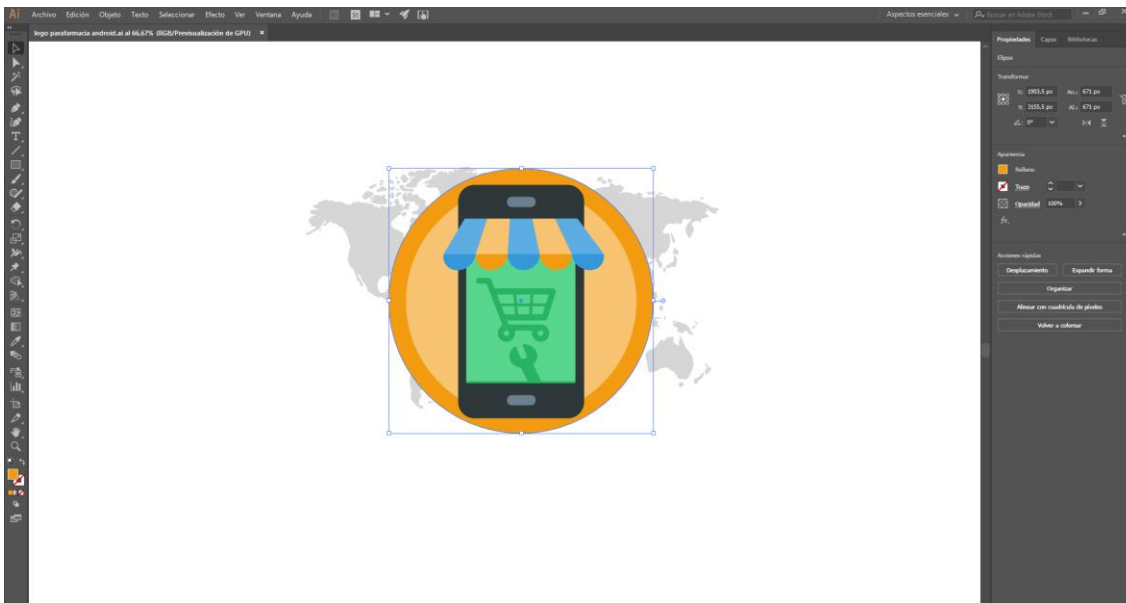


Ilustración 57 Pantalla de ejemplo de Illustrator

La funcionalidad principal del programa ha sido la de diseñar y realizar alguno de los logos e imágenes utilizadas en el proyecto.

7.3.5 GitHub

GitHub es un repositorio que permite alojar proyectos utilizando un sistema de control de versiones. Principalmente se utiliza para crear código fuente público que puede ser actualizado o mejorado por el resto de usuarios. El código fuente puede estar escrito en cualquier lenguaje. Además, cuenta con una versión de pago para almacenar proyectos de forma privada, comúnmente utilizado por empresas para tener un sistema de control de versiones en sus proyectos y facilitar la edición simultánea de varios desarrolladores.

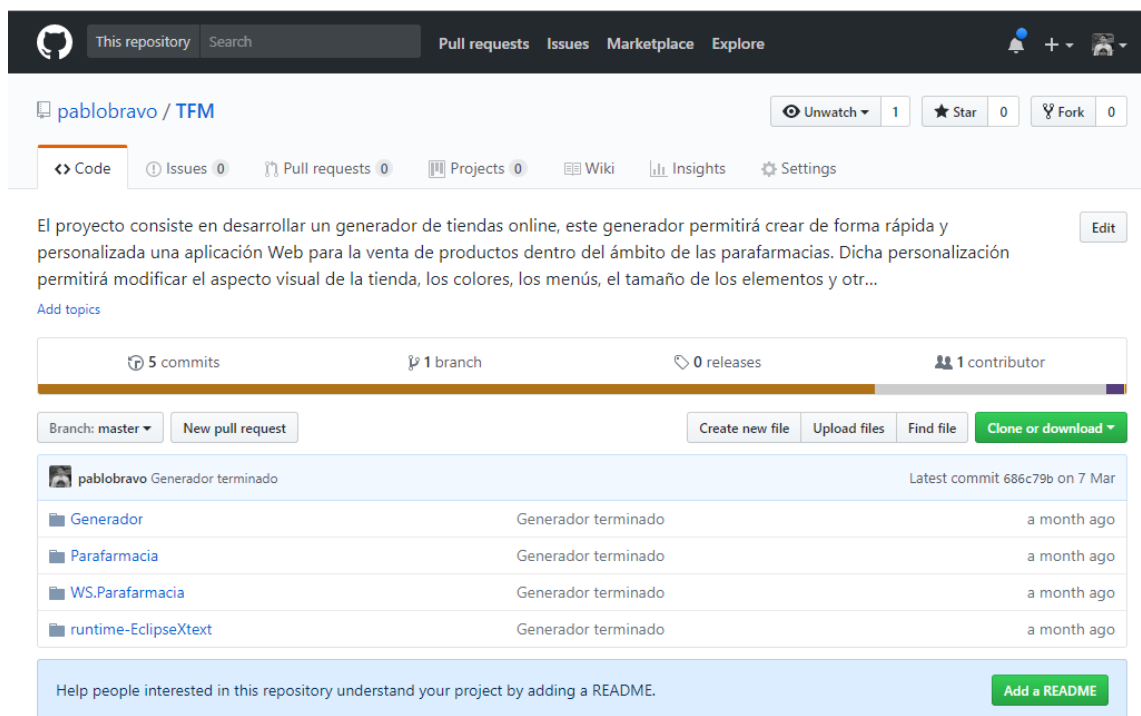


Ilustración 58 Página principal GitHub con el proyecto

En este proyecto se ha utilizado GitHub ir guardando todas las partes del proyecto que se iban desarrollando, lo que ha sido útil para tener el proyecto a salvo, como una copia de seguridad y para controlar los cambios que se iban desarrollando de una versión a otra.

7.3.6 PHPMyAdmin

Para gestionar la base de datos se ha utilizado PHPMyAdmin, que es una herramienta desarrollada en PHP que permite administrar las bases de datos MySQL a través de páginas web. Con esta herramienta podemos crear y eliminar tablas, así como sus campos y los datos que se van a guardar en ellas.

Otra cosa que podemos hacer con esta herramienta es exportar las tablas con los datos que hay en ellas de manera cómoda, lo que viene muy bien para este proyecto, ya que para cada parafarmacia generada debemos crear su propia base de datos con sus tablas y datos, de esta forma podemos transportar esta base de datos al servidor donde se tenga que alojar la nueva parafarmacia web.

7.3.7 Draw.io

Draw es una herramienta gratuita, disponible *online*, sin necesidad de descargarla, que se utiliza para construir todo tipo de diagramas, *entidad-relación*, UML (Lenguaje unificado de modelado) o BPMN (Business Process Model and Notation).

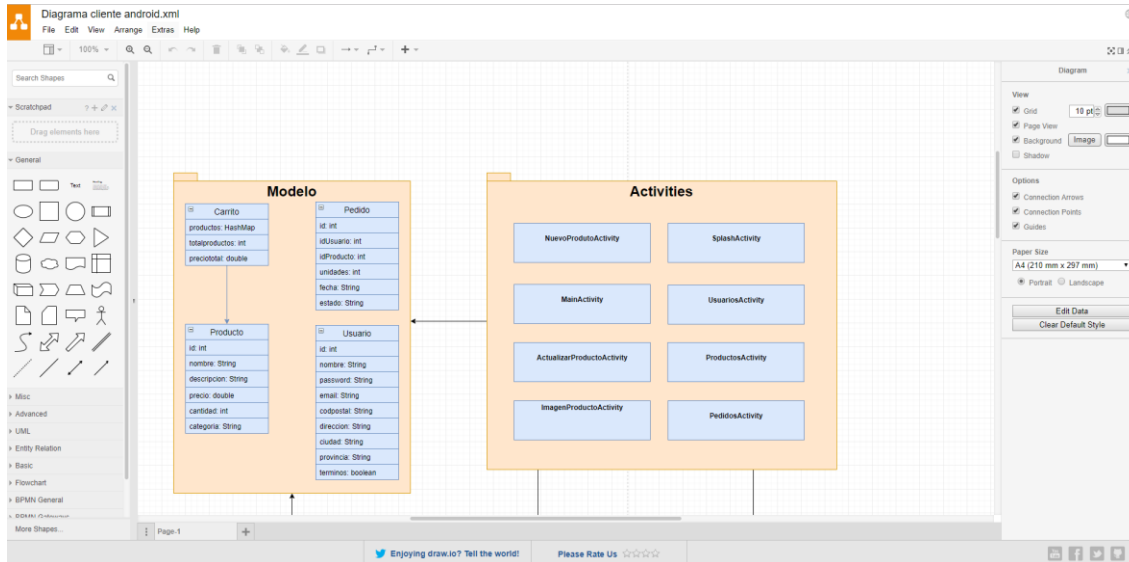


Ilustración 59 Herramienta Draw con el diagrama de paquetes del cliente Android

En este proyecto, hemos utilizado esta herramienta para construir todos los diagramas contenidos en esta documentación, arquitectura general, diagramas de paquetes, de clases...

7.3.8 Xampp

Xampp es un paquete que incluye un conjunto de software libre para facilitar el despliegue y las pruebas de aplicaciones web. Entre otras cosas, tenemos un sistema de gestión de bases de datos MySQL que será utilizado para almacenar los datos de la parafarmacia y un servidor apache, que además de alojar la base de datos, soporta el despliegue de cualquier tipo de aplicación desarrollada con PHP. Este servidor se ha utilizado para alojar código PHP que permite la subida de imágenes a la parafarmacia.

7.3.9 XText

Xtext es un *framework* para la creación de lenguajes de dominio específico creado por Itemis y que está albergado dentro de la plataforma Eclipse. Su primera versión fue publicada en el año 2006, a partir de ahí ha ido evolucionando junto con las versiones que se publicaban de Eclipse.

Mediante esta herramienta podemos definir un lenguaje personalizado ya que soporta toda la infraestructura necesaria para ello, los compiladores, intérpretes y editores. Además, permite añadir a nuestro lenguaje validaciones, warnings, errores, iconos, imágenes y autocompletar.

Para el desarrollo de mi proyecto, he decidido utilizar este *framework* con la última versión disponible actualmente. La he utilizado junto con el entorno Eclipse, para realizar el generador de código de las parafarmacias. Mediante un lenguaje de dominio creado para dicho fin, se generan una serie de archivos JSP y CSS que componen la parte visible de la parafarmacia web que se pretende generar.

```

grammar org.xtext.dsl.generador.Generador with org.eclipse.xtext.common.Terminals

generate generador "http://www.xtext.org/dsl/generador/Generador"

Parafarmacia:
  Parafarmacia '{'
    '<' Nombre '=' titulo=STRING ':' fuente=STRING '>'
    '<' Titular '=' nombre=STRING ':' direccion=Direccion ':' cif=STRING ':' registro=Registro
      ':' correo=Correo ':' telefono=INT '>'
    '<' Menu '=' tipo=Tipo '>'
    '<' Apariencia '=' apariencia=Apariencia '>'
    '<' Categorias '=' categorias+=Categoria* '>'
    '<' Productos '=' productosPorFila ':' numProductosPorFila=INT productos+=Producto '>'
    '<' Usuario '=' nombreUsuario=STRING ':' email=Correo ':' password=STRING '>'
  '}'
;

Apariencia:
  '[' Color '=' 'principal' ':' principal=Color ':' 'principalClaro' ':' principalClaro=Color ':'
    'secundario' ':' secundario=Color ':' 'fondo' ':' fondo=Color ':' 'fondoProductos' ':' fondoProductos=Color ':' 'precios' ':' precios=Color
    ':' 'texto' ':' texto=Color ':' 'titulo' ':' titulo=Color ':'
  '[' Texto '=' 'normal' ':' normal=Medida ':' 'grande' ':' grande=Medida ':'
    'titulo' ':' titulo=Medida ':' 'menu' ':' menu=Medida ':'
  '[' Nangen '=' lateral=Medida ':'
;

```

Ilustración 60 Ejemplo de DSL del generador con Xtext

7.4 Creación del Sistema

7.4.1 Problemas Encontrados

7.4.1.1 Carga de imágenes

En el subsistema del cliente Android, se muestra una lista de los productos que tiene la parafarmacia web junto a sus imágenes. Estas imágenes se obtienen de la URL en la que se encuentran alojadas, por ello, cuando se accede a la pantalla de productos, la aplicación se volvía mucho más lenta, tardaba varios segundos en cargarse y el desplazamiento por la lista de productos era a saltos.

La solución a este problema la encontré durante la exposición del proyecto en la empresa “The cocktail”, en la que trabajo actualmente, uno de los presentes me sugirió el uso de la caché de la aplicación para almacenar las imágenes. Tras esto, me puse a investigar acerca de librerías que pudieran hacer el trabajo por mí, encontrando la librería “Picasso” que con una simple línea de código carga las imágenes según se va avanzando por la lista, así que se redujo el tiempo de carga a prácticamente nada. Además, con la opción de cachear imágenes, si accedemos a la aplicación sin disponer de una conexión a internet, podemos ver que las imágenes se muestran igualmente, esto, también permite ahorrarse datos móviles ya que no se cargan todas las imágenes en cada acceso como pasaba al principio.

7.4.1.2 Añadir imágenes

Desde el cliente Android, se quiere tener la opción de modificar o añadir imágenes a los productos, la primera alternativa para conseguir esto era almacenar las imágenes en la base de datos a través de servicios web. Después de probarlo comprobé que las imágenes perdían mucha calidad y consultando con algún experto me confirmaron que debía olvidarme de esa idea.

La solución final, es la creación de un servicio escrito con lenguaje PHP que permite insertar imágenes directamente en el directorio de la parafarmacia web. Este servicio se implementa en el cliente Android mediante un WebView y una larga y complicada configuración para conseguir su funcionamiento en el mayor número de versiones Android.

7.4.1.3 Generar automáticamente la parafarmacia Web

Tras desarrollar el generador, el resultado era poco visible, ya que lo único que hacía era crear los archivos necesarios para formar la parafarmacia web, pero se guardaban dentro del propio generador, así que era necesario coger los archivos generados y copiarlos a sus respectivas carpetas.

La solución a este problema fue, en primer lugar, crear un archivo de configuración el que se escribe la ruta en la que se encuentra la parafarmacia web, de esta forma, los archivos generados se copian automáticamente en el lugar correspondiente. Esto no era suficiente ya que para tener funcionando la nueva parafarmacia había que actualizar y compilar el proyecto de nuevo, así que el segundo paso fue cambiar la configuración del proyecto para que se actualice y compile solo cada vez que recibe un nuevo archivo.

7.4.1.4 Múltiples conexiones HSQLDB

En el inicio del proyecto, se decidió utilizar una base de datos HSQLDB, como se había diseñado, la parafarmacia web iba a conectarse con ella de forma directa mientras que el resto de subsistemas iban a hacer uso de los servicios web. Esto hace un total de dos conexiones simultáneas a la base de datos. Después de muchas pruebas, no fue posible que ambas conexiones funcionaran a la vez. Tras una larga investigación, apareció el problema, la base de datos no permitía dicho acceso simultáneo.

La solución a este problema fue cambiar la base de datos utilizada, ya que MySQL si permitía múltiples conexiones se decidió usar esta en sustitución de la anterior. Como ambos subproyectos disponían de un archivo de configuración para conectarse con las bases de datos, el problema no se extendió mucho ya que solo fue necesario cambiar los drivers y las credenciales para que los proyectos funcionaran a la perfección, junto a la creación de las tablas en la nueva base de datos.

7.4.2 Descripción Detallada de las Clases

7.4.2.1 Parafarmacia Web

7.4.2.1.1 Producto

Nombre	Tipo	Descripción	Hereda de...
Producto	Clase	Esta clase forma parte del modelo y se utilizará por las clases encargadas de acceder a la base de datos para insertar, obtener, actualizar o eliminar productos	
Métodos			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público		Getter y Setter de todos los atributos	
Atributos			
Acceso	Modo	Tipo o Clase	Nombre
privado		Int	Cantidad
privado		String	Categoría
privado		String	Descripción
privado		Int	Id
privado		String	Nombre
privado		Double	Precio

Tabla 22 Clase producto parafarmacia web

7.4.2.1.2 Pedido

Nombre	Tipo	Descripción	Hereda de...
Pedido	Clase	Esta clase forma parte del modelo y se utilizará por las clases encargadas de acceder a la base de datos para insertar, obtener, actualizar o eliminar pedidos	
Métodos			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público		Getter y Setter de todos los atributos	
Atributos			
Acceso	Modo	Tipo o Clase	Nombre
privado		Int	Id
privado		String	Estado
privado		String	Fecha
privado		Int	IdProducto
privado		Int	IdUsuario
privado		Int	Unidades

Tabla 23 Clase pedido parafarmacia web

7.4.2.1.3 Carrito

Nombre	Tipo	Descripción	Hereda de...
Carrito	Clase	Esta clase forma parte del modelo y se utilizará por los controladores para almacenar en sesión el carrito de la compra del usuario con los productos añadidos a él	
Métodos			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público		Getter y Setter de todos los atributos	
público		AddProductos	Producto
público		RemoveProductos	Producto
Atributos			
Acceso	Modo	Tipo o Clase	Nombre
privado		Double	PrecioTotal
privado		HashMap<Producto>	Productos
privado		Int	TotalProductos

Tabla 24 Clase carrito parafarmacia web

7.4.2.1.4 Usuario

Nombre	Tipo	Descripción	Hereda de...
Usuario	Clase	Esta clase forma parte del modelo y se utilizará por las clases encargadas de acceder a la base de datos para insertar, obtener, actualizar o eliminar usuarios	
Métodos			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público		Getter y Setter de todos los atributos	
Atributos			
Acceso	Modo	Tipo o Clase	Nombre
privado		Int	Id
privado		String	Ciudad
privado		String	Codpostal
privado		String	Dirección
privado		String	Provincia
privado		String	Email
privado		String	Nombre
privado		String	Password
privado		String	PasswordRepeat

Tabla 25 Clase usuario parafarmacia web

7.4.2.1.5 Pedido DAO

Nombre	Tipo	Descripción	Hereda de...
PedidoDAO	Clase	Esta clase se encarga de conectarse con la base de datos mediante el uso de archivos de configuración para añadir, obtener y actualizar pedidos	PedidoDataService
<i>Métodos</i>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público	Vector<Pedido>	GetPedidos	
público		NewPedido	Pedido
público		UpdatePedido	Pedido

Tabla 26 Clase pedido DAO parafarmacia web

7.4.2.1.6 Producto DAO

Nombre	Tipo	Descripción	Hereda de...
ProductoDAO	Clase	Esta clase se encarga de conectarse con la base de datos mediante el uso de archivos de configuración para añadir, eliminar, obtener y actualizar productos	ProductoDataService
<i>Métodos</i>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público	Vector<Producto>	GetProductos	
público		NewProducto	Pedido
público		UpdateProducto	Pedido
público		DeleteProducto	

Tabla 27 Clase producto DAO parafarmacia web

7.4.2.1.7 Usuario DAO

Nombre	Tipo	Descripción	Hereda de...
UsuarioDAO	Clase	Esta clase se encarga de conectarse con la base de datos mediante el uso de archivos de configuración para añadir, obtener y actualizar usuarios	UsuarioDataService
<i>Métodos</i>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público	Vector<Usuario>	GetUsuarios	
público		NewUsuario	Usuario
público		UpdateUsuario	Usuario

Tabla 28 Clase usuario DAO parafarmacia web

7.4.2.1.8 WelcomeController

Nombre	Tipo	Descripción	Hereda de...
WelcomeController	Clase	Esta clase es un controlador encargado de pedir los datos necesarios y enviarlos a la vista solicitada por el usuario.	Controller
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público	String	Inicio	Request, Model
público	String	Welcome	Request, Model

Tabla 29 Clase welcome controller parafarmacia web

7.4.2.1.9 AccesoController

Nombre	Tipo	Descripción	Hereda de...
AccesoController	Clase	Esta clase se encarga de recibir los datos introducidos por el usuario para el inicio de sesión, comprobar que los datos existen y son correctos y si así fuera, guarda el usuario en sesión	Controller
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público	Usuario	getUsuario	
público	String	IniciarSesión	Request, Model, Usuario

Tabla 30 Clase acceso controller parafarmacia web

7.4.2.1.10 ActualizarCarritoController

Nombre	Tipo	Descripción	Hereda de...
ActualizarCarritoController	Clase	Esta clase se encarga de gestionar el carrito del usuario durante su compra, permite añadir o eliminar productos de él	Controller
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público	String	addProductoCarrito	Request, Model, Carrito
público	String	RemoveProductoCarrito	Request, Model, Carrito

Tabla 31 Clase actualizar carrito controller parafarmacia web

7.4.2.1.11 BuscadorController

Nombre	Tipo	Descripción	Hereda de...
BuscadorController	Clase	Esta clase se encarga de filtrar los productos de la parafarmacia según el texto pasado por el usuario	Controller
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público	String	buscar	Request, Model, Buscar

Tabla 32 Clase buscador controller parafarmacia web

7.4.2.1.12 CarritoController

Nombre	Tipo	Descripción	Hereda de...
CarritoController	Clase	Esta clase se encarga de añadir a la sesión el carrito para que se encuentre a disposición del usuario	Controller
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público	String	Carrito	Model

Tabla 33 Clase carrito controller parafarmacia web

7.4.2.1.13 CuentaController

Nombre	Tipo	Descripción	Hereda de...
CuentaController	Clase	Esta clase se encarga de la gestión de la cuenta del usuario con sesión iniciada en el sistema, permite actualizar los datos	Controller
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público	String	modificarCuenta	Request, Model, Usuario

Tabla 34 Clase cuenta controller parafarmacia web

7.4.2.1.14 DetallesController

Nombre	Tipo	Descripción	Hereda de...
DetallesController	Clase	Esta clase se encarga de obtener los datos de un producto y enviarlos a la vista para que se muestren al usuario	Controller
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público	String	detalles	Request, Model

Tabla 35 Clase detalles controller parafarmacia web

7.4.2.1.15 FinalizarPedidoController

Nombre	Tipo	Descripción	Hereda de...
FinalizarPedidoController	Clase	Esta clase se encarga de obtener los datos del pedido realizado por el usuario y enviarlos a la clase que lo añadirá en la base de datos	Controller
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público	String	finalizarPedido	Request, Model

Tabla 36 Clase finalizar pedido controller parafarmacia web

7.4.2.1.16 PedidosController

Nombre	Tipo	Descripción	Hereda de...
PedidosController	Clase	Esta clase se encarga de obtener los datos de todos los pedidos realizados por el usuario con sesión iniciada y enviarlos a la vista para que se muestren	Controller
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público	String	misPedidos	Request, Model

Tabla 37 Clase pedidos controller parafarmacia web

7.4.2.1.17 ProductosController

Nombre	Tipo	Descripción	Hereda de...
ProductosController	Clase	Esta clase se encarga de obtener todos los productos de la base de datos a través de la clase correspondiente y enviar dichos productos a la vista para que sean mostrados al usuario	Controller
<i>Métodos</i>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público	String	productos	Request, Model
público	String	Bebes	Request, Model
público	String	Cosmética	Request, Model
público	String	Dietética	Request, Model
público	String	Higiene	Request, Model
público	String	Salud	Request, Model

Tabla 38 Clase productos controller parafarmacia web

7.4.2.2 Cliente Android

7.4.2.2.1 SplashActivity

Nombre	Tipo	Descripción	Hereda de...
SplashActivity	Clase	Clase que funciona como un controlador que se encarga de inicializar la vista principal de la aplicación durante cinco segundos y a continuación instanciar una nueva clase. Además, durante los segundos que se muestra, realiza una conexión con el servicio web para obtener todos los datos almacenados en la base de datos y guardarlos en su propia base de datos	AppCompatActivity
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público		getAll	
público		parsePedidos	String
público		parseProductos	String
público		parseUsuarios	String
<u>Atributos</u>			
Acceso	Modo	Tipo o Clase	Nombre
privado	final	String	NAMESPACE
privado	final	String	URL
privado	final	String	SOAP_ACTION
privado	final	String	METHOD_NAME
privado		String	resultadoPedidos
privado		String	resultadoProductos
privado		String	resultadoUsuarios
privado		List	pedidos
privado		List	usuarios
privado		List	productos

Tabla 39 Clase splash activity cliente Android

7.4.2.2.2 ProductosActivity

Nombre	Tipo	Descripción	Hereda de...
ProductosActivity	Clase	Clase encargada de obtener los productos y enviarlos a la vista para que sean mostrados al usuario, también añade la opción de actualizar o crear productos	AppCompatActivity
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público		nuevoProducto	View
público		actualizarImagen	View
público		actualizarProducto	View
<u>Atributos</u>			
Acceso	Modo	Tipo o Clase	Nombre
privado	final	ListView	IvProductos
privado	final	ProductosArrayAdapter	currentAdaptador
privado	final	List	productosBD

Tabla 40 Clase productos activity cliente Android

7.4.2.2.3 PedidosActivity

Nombre	Tipo	Descripción	Hereda de...
PedidosActivity	Clase	Clase encargada de obtener los pedidos y enviarlos a la vista para que sean mostrados al usuario, también da la opción de actualizar el estado de los pedidos	AppCompatActivity
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público		actualizarPedido	View
<u>Atributos</u>			
Acceso	Modo	Tipo o Clase	Nombre
privado	final	ListView	IvPedidos
privado	final	PedidoArrayAdapter	currentAdaptador

Tabla 41 Clase pedidos activity cliente Android

7.4.2.2.4 UsuariosActivity

Nombre	Tipo	Descripción	Hereda de...
UsuariosActivity	Clase	Clase encargada de obtener los usuarios y enviarlos a la vista para que se muestren al usuario	AppCompatActivity
<u>Atributos</u>			
Acceso	Modo	Tipo o Clase	Nombre
privado	final	ListView	IvUsuarios

Tabla 42 Clase usuarios activity cliente Android

7.4.2.2.5 MyDBHelper

Nombre	Tipo	Descripción	Hereda de...
MyDBHelper	Clase	Clase encargada de crear las tablas de la base de datos SQLite con todos los campos indicados	SQLiteOpenHelper
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público		onCreate	SQLiteDatabase
público		onUpgrade	SQLiteDatabase, int
<u>Atributos</u>			
Acceso	Modo	Tipo o Clase	Nombre
privado	final	String	Tabla_usuarios
privado	final	String	Tabla_productos
privado	final	String	Tabla_pedidos
privado	final	String	atributos
privado	final	String	BD_create_pedidos
privado	final	String	BD_create_productos
privado	final	String	BD_create_usuarios

Tabla 43 Clase my DB helper cliente Android

7.4.2.2.6 PedidosDataSource

Nombre	Tipo	Descripción	Hereda de...
PedidosDataSource	Clase	Clase encargada de conectarse a la base de datos para insertar, obtener o actualizar los pedidos	
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público		insertPedido	Pedido
público	List	getPedidos	
público		updateEstadoPedidos	Estado, id, fecha
público		deleteAllPedidos	
<u>Atributos</u>			
Acceso	Modo	Tipo o Clase	Nombre
privado		SQLiteDatabase	database
privado		MyDBHelper	dbHelper

Tabla 44 Clase pedidos data source cliente Android

7.4.2.2.7 ProductosDataSource

Nombre	Tipo	Descripción	Hereda de...
ProductosDataSource	Clase	Clase encargada de conectarse a la base de datos para insertar, obtener o actualizar los productos	
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público		insertProducto	Producto
público	List	getProductos	
público		updateProducto	Producto
público		getProducto	id
<u>Atributos</u>			
Acceso	Modo	Tipo o Clase	Nombre
privado		SQLiteDatabase	database
privado		MyDBHelper	dbHelper

Tabla 45 Clase productos data source cliente Android

7.4.2.2.8 UsuariosDataSource

Nombre	Tipo	Descripción	Hereda de...
UsuariosDataSource	Clase	Clase encargada de conectarse a la base de datos para insertar, obtener o actualizar los usuarios	
<u>Métodos</u>			
Acceso Modo	Tipo de Retorno	Nombre	Parámetros y tipos
público		insertUsuario	Usuario
público	List	getUsuarios	
público		getUsuario	id
<u>Atributos</u>			
Acceso	Modo	Tipo o Clase	Nombre
privado		SQLiteDatabase	database
privado		MyDBHelper	dbHelper

Tabla 46 Clase usuarios data source cliente Android

Capítulo 8. Desarrollo de las Pruebas

8.1 Pruebas Unitarias

8.1.1 Base de datos

Las siguientes pruebas realizadas son iguales para tres de los subsistemas del proyecto que tienen clases para acceder a la base de datos, la parafarmacia web, los servicios web y el cliente Android.

Base de datos			
Resultado Obtenido			
Prueba	Parafarmacia Web	Servicio Web	Cliente Android
Obtener usuarios: esta prueba consiste en realizar una consulta de obtención de todos los usuarios que contiene la base de datos y comprobar que el número de usuarios es correcto.			
Insertar usuario: esta prueba consiste en añadir un nuevo usuario a la base de datos y comprobar que se ha añadido correctamente.			No disponible
Actualizar usuario: esta prueba consiste en actualizar los datos de un usuario y comprobar que se han actualizado correctamente.			No disponible
Obtener productos: esta prueba consiste en realizar una consulta de obtención de todos los usuarios que contiene la base de datos y comprobar que el número de productos es correcto.			
Insertar producto: esta prueba consiste en añadir un nuevo producto a la base de datos y comprobar que se ha añadido correctamente.			

Borrar producto: esta prueba consiste en borrar un producto de la base de datos y comprobar que la cantidad de productos se ha disminuido en una y el producto eliminado no está.			
Actualizar producto: esta prueba consiste en actualizar los datos de un producto y comprobar que se han actualizado correctamente.			
Insertar pedido: esta prueba consiste en añadir un nuevo pedido a la base de datos y comprobar que se ha añadido correctamente.			No disponible
Actualizar pedido: esta prueba consiste en actualizar los datos de un pedido y comprobar que se han actualizado correctamente.			
Obtener pedidos: esta prueba consiste en realizar una consulta de obtención de todos los pedidos que contiene la base de datos y comprobar que el número de pedidos es correcto.			

Tabla 47 Resultado pruebas unitarias base de datos

8.1.2 Generador

Para realizar las pruebas del generador, escribimos un código de generación y comprobamos que los archivos generados contienen los datos introducidos en el código.

Código de generación:

```

Parafarmacia {
  < Nombre = "BravoFarma" : "Arial" >
  < Titular = "Pablo" : [Direccion = "calle estación" : "Colloto" : "España" ] :
    "345632" : [Registro = 19 : 2 : 5 : 28] : pablo@gmail.com : 665666944
  >
  < Menu = Permanente : #333: 60px >
  < Apariencia =
    [Color = principal.#408733 : principalClaro.#CEF6CE : secundario.#89c441
  : fondo.#e7e8ea : fondoProductos.#fff : precios.#FE642E : texto.#2E2E2E :
  titulo.#585858]
    [Texto = normal.14px : grande.16px : titulo.26px : menu.20px]
    [Margen = 10%]
  >
  < Categorías = Salud Cosmetica Higiene Dietetica Bebes >
  < Productos = productosPorFila.3>
  < Usuario = "pablo" : pablo@gmail.com : "pablo" >
}
    
```

Comprobamos el primer archivo generado, style.css:

```
:root {  
    /* Colores */  
    --main-color: #408733;  
    --main-color-claro: #CEF6CE;  
    --second-color: #89c441;  
    --nav-color:#333;  
    --background-color:#e7e8ea;  
    --color-background-prductos: #fff;  
    --color-text-nav: #fff;  
    --color-precios: #FE642E;  
    --color-text: #2E2E2E;  
    --color-titulo: #585858;  
  
    /* Tamaños elementos */  
    --margenes-laterales: 10%;  
    --altura-menu: 60px;  
  
    --ancho-productos:calc(33% - 50px);  
  
    /* Tamaños textos */  
    --size-texto: 14px;  
    --size-texto-grande:16px;  
    --size-menu:20px;  
    --size-titulo:26px;  
  
    /* Familia fuentes */  
    --familia-texto: Helvetica,Arial sans-serif;  
    --familia-titulos:serif,'Alegreya Sans SC' ;  
}
```

Como se puede comprobar en este trozo de código, los códigos de los colores creados son los mismos introducidos en el código de generación. De esta forma se ha comprobado que el código generado sea igual al código esperado.

Esto mismo se realiza para el resto de archivos generados, por ejemplo, [*inicio.jsp*](#):

```
<div class="categoria"><a href="salud"><spring:message code="salud" /><img src='<c:url  
value='./resources/categorias/categoria3.jpg'/>' alt='<spring:message code="salud"  
>'></a></div>  
    <div class="categoria"><a  
href="cosmetica"><spring:message code="cosmetica" /><img src='<c:url  
value='./resources/categorias/categorial.jpg'/>' alt='<spring:message code="cosmetica"  
>'></a></div>  
    <div class="categoria"><a  
href="higiene"><spring:message code="higiene" /><img src='<c:url  
value='./resources/categorias/categoria2.jpg'/>' alt='<spring:message code="higiene"  
>'></a></div>  
    <div class="categoria"><a  
href="dietetica"><spring:message code="dietetica" /><img src='<c:url  
value='./resources/categorias/categoria5.jpg'/>' alt='<spring:message code="dietetica"  
>'></a></div>  
    <div class="categoria"><a  
href="bebes"><spring:message code="bebes" /><img src='<c:url  
value='./resources/categorias/categoria4.jpg'/>' alt='<spring:message code="bebes"  
>'></a></div>
```

Se puede ver en el trozo de código anterior que se han incluido las categorías de productos en el apartado correspondiente.

8.2 Pruebas de Integración y del Sistema

8.2.1 Generador - servicio Web

Para realizar las pruebas de integración escribimos el código de generación de productos, con lo que el generador hará uso del servicio web e insertará los productos en la base de datos de la parafarmacia web. Código de generación:

```
< Productos = productosPorFila.3
  [Producto = 'ACOFARDERM COLUTORIO ACCIÓN TOTAL': 'Si quieres una higiene bucal completa, no basta con cepillarse los dientes después de cada comida. La placa bacteriana puede adherirse a nuestro esmalte con el paso del tiempo, y un cepillado constante no basta para hacerla desaparecer, sobre todo en los espacios interdientales o más inaccesibles.\n\nCon Acofardent colutorio acción total aportarás a tu higiene bucal una acción anticaries y antiplaca completa que te ayudará a reducir el riesgo de caries y a eliminar más placa bacteriana.\n\nAdemás, su fórmula rica en alantoína te ayudará a proteger la mucosa dental y a fortalecer tus encías.': 33.98: 1: Higiene]
  [Producto = 'ACOFARDERM BÁLSAMO REPARADOR NARIZ Y LABIOS': 'Con la llegada del invierno, es muy fácil coger un resfriado que te deje la nariz pelada. También se produce este tipo de irritaciones cuando se padece una alergia. Si lo que buscas es un bálsamo para aliviar la irritación causada al sonarse la nariz, este tratamiento de dermatocósmética te irá de perlas.\n\nAcofarderm Bálsamo Reparador hidrata los labios secos y sensibles y los protege frente a las radiaciones solares y el frío, a nivel SPF10+. Un protector labial para labios secos que ha sido testado dermatológicamente. Contiene Aloe Vera, que hidrata y refresca la zona, y Vitamina E, que produce una acción antioxidante.\n\nTus labios secos y sensibles lucirán más hidratados que nunca y además estarán protegidos de las radiaciones solares. Tu nariz, dejará de estar enrojecida y ya no te molestará.': 22.48: 150: Cosmetica]
  [Producto = 'ACOFARDERM CREMA DE MANOS CONCENTRADA': 'Si trabajas al aire libre o en condiciones ambientales adversas, utilizar una crema nutritiva es fundamental para mantener en buen estado tus manos. En esas condiciones, es normal que éstas se estropeen y terminen mostrando un aspecto apagado, tirante y reseco. Cuidalas y nutrelas a diario para evitar que pierdan vitalidad y belleza.\n\nLa crema de manos concentrada de Acofarderm es una solución ideal para las manos más secas y estropeadas. Esta crema altamente hidratante, rica en coenzima Q10 y glicerina, le aportará a tus manos una sensación de alivio inmediato y duradero, además de nutrir las en profundidad. Crema apta incluso para manos en condiciones adversas, extremas o límites.': 2.2: 98: Higiene]
  [Producto = 'ACOFARDERM CREMA DE MANOS CUIDADO DIARIO 75ML': 'La marca Acofarderm es especialista en cuidado facial, elaborando productos altamente eficaces para proteger la piel de los agentes externos. Te recomendamos dos productos imprescindibles en tu rutina de belleza diaria, sobre todo si tienes la piel seca.\n\nEste pack hidratante de Acofarderm contiene:\n\n- Acofarderm crema de manos 75ml: Repara e hidrata intensamente la piel de las manos más secas o agrietadas. Ideal para adultos con la piel seca o estropeada por causa del medioambiente (frío, calor, contaminación...). Se trata de una crema ultrahidratante elaborada con urea y aloe vera, dos componentes activos con propiedades hidratantes y antiinflamatorias. Su textura suave y absorbente ayuda a dejar la piel lisa y sedosa. \n\n- Acofarderm bálsamo labial SPF15+ 4g: El stick labial de Acofarderm protege la piel de los labios de cualquier agente externo (radiaciones solares, frío, viento, contaminación...). Ideal para personas con los labios secos, irritados o agrietados. Además, proporciona la máxima elasticidad para tu piel. ¡Con protección solar SPF15+!': 1.65: 150: Higiene]
  [Producto = 'ACOFARDERM GEL DERMATOLÓGICO': 'Cada piel necesita un cuidado específico, por ello necesitamos prestar especial atención a los productos que aplicamos a diario sobre ella. Los principios activos de Acofarderm te ayudan a mantener tu piel totalmente reparada y saludable en todo momento.\n\nAcofarderm gel dermatológico 0% es un delicado limpiador, formulado con menor cantidad de ingredientes de origen químico. Limpia la piel en profundidad y deja un tacto agradable y con mayor suavidad.\n\nSu formulación sin parabenos ni colorantes, trata tu piel con la delicadeza y protección que necesita. Puede ser usado por todo tipo de piel, debido a su elevada tolerancia cutánea.': 1.98: 98: Higiene]
  [Producto = 'ACOFARDERM LOCIÓN CORPORAL': 'Algunas personas sufren de piel atópica, una enfermedad de la piel que causa la inflamación crónica de la misma, especialmente en las zonas del cuerpo que están siempre en movimiento como son codos, rodillas o el propio cuero cabelludo. Se trata de una de las enfermedades cutáneas más comunes, muy frecuente en los niños, y que exige un cuidado especial de la piel.\n\nLa loción corporal para pieles atópicas de Acofarderm está indicada para las personas que sufren este trastorno de la piel, aportando activos hidratantes y protectores pero sin incluir parabenos que podrían acabar causando un mayor trastorno de la piel.\n\nLa loción para pieles atópicas de Acofarderm ayuda a reducir la formación de rojeces o
```

```

irritaciones de la piel, llegando a aparecer en ocasiones escamas debido a la
hipersecreción de grasa prurito.': 3.48: 150: Higiene]
[Producto = 'ACOFAR REPELENTE DE INSECTOS': 'En épocas de excursiones,
viajes y campamentos, es esencial contar con productos repelentes de insectos, que te
ayuden a proteger a tu familia de sus picaduras. Especialmente por el calor y la
humedad, aparecen los mosquitos y otros insectos, que producen molestias en nuestro
cuerpo. \n\nEl spray repelente insectos de Acofar es un producto con un efecto de hasta
8 horas de duración capaz de ahuyentar todo tipo de insectos, incluidos los mosquitos
tigres. Gracias a una barrera protectora degradable olor, evitarás las picaduras y
ronchones de los insectos. Apto tanto para adultos como para niños.': 4.48: 150: Salud ]
[Producto = 'A DERMA AVOINE RHEALBA': 'Los dermatólogos afirman que las
pieles con tendencia a piel atópica requieren un cuidado limpiador adaptado. El aceite
de baño Exomega de A-derma es una de las soluciones para la higiene diaria de las pieles
atópicas y muy secas tanto en bebé, niño, como adulto.\n\nSu textura emoliente y sedosa
y su composición a base de plántulas de Avena Rhealba® hacen que sea un aceite adaptado
para las pieles más secas. ': 13.48: 150: Cosmetica ]
[Producto = 'ACOFARDERM LECHE LIMPIADORA TÓNICO': 'Desmaquillarse es
fundamental para conseguir una piel sana, brillante y luminosa. Uno de los riesgos que
corremos al no desmaquillarnos es perder luminosidad en nuestra piel, provocando una
sensación de piel apagada y cansada. Otro de los peligros es la aparición de alergias a
determinados productos o a aparición de los nada deseados granitos de acné. Y la última
consecuencia y más dañina es el envejecimiento prematuro de la piel.\n\nDesmaquillarse
ahora es mucho más fácil gracias a Acofarderm leche limpiadora + tónico, dos productos
en uno. Su suave textura permite eliminar suavemente todos los restos de maquillaje e
impurezas de la piel, mientras tonifica el rostro.': 5.48: 15: Cosmetica]
>
    
```

El resultado obtenido es el siguiente:

+ Opciones						
	id	nombre	descripcion	precio	cantidad	categoria
<input type="checkbox"/>	1	ACOFARDENT COLUTORIO ACCIÓN TOTAL	Si quieres una higiene bucal completa, no basta co...	33.98	1	HIGIENE
<input type="checkbox"/>	2	ACOFARDERM BALSAMO REPARADOR NARIZ Y LABIOS	Con la llegada del invierno, es muy fácil coger un...	22.48	150	COSMÉTICA
<input type="checkbox"/>	3	ACOFARDERM CREMA DE MANOS CONCENTRADA	Si trabajas al aire libre o en condiciones ambient...	2.2	98	HIGIENE
<input type="checkbox"/>	4	ACOFARDERM CREMA DE MANOS CUIDADO DIARIO 75ML	La marca Acofarderm es especialista en cuidado fac...	1.65	150	HIGIENE
<input type="checkbox"/>	5	ACOFARDERM GEL DEMATOLÓGICO	Cada piel necesita un cuidado específico, por ello...	1.98	98	HIGIENE
<input type="checkbox"/>	6	ACOFARDERM LOCIÓN CORPORAL	Algunas personas sufren de piel atópica, una enfer...	3.48	150	HIGIENE
<input type="checkbox"/>	7	ACOFAR REPELENTE DE INSECTOS	En épocas de excursiones, viajes y campamentos, es...	4.48	150	SALUD
<input type="checkbox"/>	8	A DERMA AVOINE RHEALBA	Los dermatólogos afirman que las pieles con tenden...	13.48	150	COSMÉTICA
<input type="checkbox"/>	9	ACOFARDERM LECHE LIMPIADORA TÓNICO	Desmaquillarse es fundamental para conseguir una pi...	5.48	15	COSMÉTICA

Ilustración 61 Salida esperada base de datos MySQL

Los productos que ya estuvieran en la base de datos se tienen que borrar e insertarse los nuevos, además se debe reiniciar el id de los productos insertados a 1.

8.2.2 Cliente Android – Servicio Web







Cliente Android – Servicio Web	
Prueba	Resultado obtenido
Nuevo producto: el cliente Android se conecta con el Servicio web para crear nuevos productos en la parafarmacia web. La prueba consiste en insertar un nuevo producto.	 La base de datos de la parafarmacia web tiene un nuevo producto con los datos enviados desde el cliente Android
Actualizar producto: esta prueba consiste en actualizar los datos de un producto.	 El producto actualizado en la base de datos de la parafarmacia web tiene los nuevos datos enviados desde el cliente Android
Obtener productos: esta prueba consiste en solicitar todos los productos de la base de datos. Se comprobará tanto que el número de productos sea el correcto como los datos recibidos de cada producto.	 El cliente Android tiene una lista con todos los productos de la parafarmacia web y sus campos
Actualizar pedido: esta prueba consiste en actualizar el estado de un pedido, ya que es el único campo que se actualizará desde el cliente Android.	 El estado del pedido actualizado en la parafarmacia web cambia al indicado desde el cliente Android
Obtener pedidos: esta prueba consiste en solicitar todos los pedidos de la base de datos. Se comprobará tanto que el número de pedidos sea el correcto como los datos recibidos de cada pedido.	 El cliente Android tiene una lista con todos los pedidos de la parafarmacia web y sus campos
Obtener usuarios: esta prueba consiste en solicitar todos los usuarios de la base de datos. Se comprobará tanto que el número de usuarios sea el correcto como los datos recibidos de cada usuario.	 El cliente Android tiene una lista con todos los usuarios de la parafarmacia web y sus campos

Tabla 48 Resultado pruebas de integración cliente Android - servicio web

8.3 Pruebas de Usabilidad y Accesibilidad

8.3.1 Pruebas de Usabilidad

A partir de los cuestionarios que se diseñaron anteriormente y de los procedimientos explicados, mostramos aquí el resultado de aplicarlos sobre los usuarios empleados para las pruebas:

El número de usuarios sobre el que se han realizado las pruebas es de cinco, se procederá a incluir sus respuestas dentro de la misma tabla, se identificará a los usuarios como Usuario1, Usuario2, Usuario3, Usuario4 y Usuario5. Los usuarios elegidos para las pruebas tienen algún conocimiento sobre ordenadores o dispositivos móviles.

Preguntas generales:

¿Usa un ordenador frecuentemente?
<ol style="list-style-type: none"> 1. Todos los días: Usuario2, Usuario5 2. Varias veces a la semana: Usuario1, Usuario3, Usuario4 3. Ocasionalmente 4. Nunca o casi nunca
¿Qué tipo de actividades realiza con el ordenador?
<ol style="list-style-type: none"> 1. Es parte de mi trabajo o profesión: Usuario2, Usuario5 2. Lo uso básicamente para ocio: Usuario3, Usuario4 3. Solo empleo aplicaciones estilo Office 4. Únicamente leo el correo y navego ocasionalmente: Usuario1
¿Ha usado alguna vez software como el de esta prueba?
<ol style="list-style-type: none"> 1. Sí, he empleado software similar: Usuario1, Usuario2, Usuario3, Usuario4, Usuario5 2. No, aunque si empleo otros programas que me ayudan a realizar tareas similares 3. No, nunca
¿Qué busca Vd. Principalmente en un programa?
<ol style="list-style-type: none"> 1. Que sea fácil de usar: Usuario2, Usuario3, Usuario4, Usuario5

2. Que sea intuitivo
3. Que sea rápido
4. Que tenga todas las funciones necesarias: **Usuario1**

8.3.1.1 Parafarmacia Web

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Sabe dónde está dentro de la aplicación?	Usuario2 Usuario3 Usuario4 Usuario5	Usuario1		
¿Le resulta sencillo el uso de la aplicación?	Usuario5 Usuario2	Usuario1 Usuario3 Usuario4		
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Funciona cada tarea como Vd. Espera?	Usuario1 Usuario2 Usuario3 Usuario4 Usuario5			
¿El tiempo de respuesta de la aplicación es muy grande?				Usuario1 Usuario2 Usuario3 Usuario4 Usuario5
Calidad del Interfaz				
Aspectos gráficos	Muy	Adecuado	Poco Adecuado	Nada

	Adecuado			Adecuado
<i>El tipo y tamaño de letra es</i>	Usuario3	Usuario1 Usuario2 Usuario4 Usuario5		
<i>Los iconos e imágenes usados son</i>	Usuario3	Usuario1 Usuario2 Usuario4 Usuario5		
<i>Los colores empleados son</i>	Usuario3 Usuario4 Usuario5	Usuario1 Usuario2		
Diseño de la Interfaz		Si	No	A veces
<i>¿Le resulta fácil de usar?</i>		Usuario1 Usuario2 Usuario3 Usuario4 Usuario5		
<i>¿El diseño de las pantallas es claro y atractivo?</i>		Usuario1 Usuario2 Usuario3 Usuario4		Usuario5
<i>¿Cree que el programa está bien estructurado?</i>		Usuario1 Usuario2 Usuario3 Usuario4		

Usuario5		
Observaciones		
Uno de los usuarios se ha quejado que cuando inicia sesión estando en el carrito de la compra debería volver al carrito en lugar de a la pantalla principal.		
La mayoría de los usuarios han preguntado si no podían escoger la forma de pagar.		
Otro usuario ha preguntado por la factura al realizar la compra.		

Tabla 49 Resultado cuestionario accesibilidad con usuarios

A continuación, se evaluarán las siguientes tareas sobre los usuarios anteriores:

- *Registrarse en la web.*
- *Cerrar sesión en la web.*
- *Iniciar sesión en la web.*
- *Ver la descripción de un producto.*
- *Realizar la compra de dos productos de la tienda.*
- *Visualizar la política de privacidad de la tienda.*

8.3.1.1.1 Usuario1

- **¿Qué tipo de usuario está realizando las tareas?**

Usuario medio-bajo que utiliza los dispositivos de vez en cuando para navegar.

- **¿El usuario comienza a realizar las tareas de forma inmediata?**

La primera tarea, “registrarse en la web”, tarda unos segundos en iniciar mientras se sitúa y busca la opción.

La tarea para visualizar la política de privacidad también tarda más de la cuenta en encontrar la opción, ya que no es un usuario familiarizado con ver estos apartados.

- **¿El usuario ha cometido algún error al realizar alguna tarea? ¿Cuál?**

No ha cometido errores.

- **¿Cuánto tiempo ha tardado el usuario en realizar cada una de las tareas?**

El registro ha sido de las tareas que más ha tardado, ya que debía escribir datos, por lo que el tiempo es de 1 minuto y 25 segundos, las dos siguientes ha sido un total de 40 segundos, tardando la mayor parte en escribir sus datos para iniciar sesión, ver la descripción de un producto fueron 10 segundos, realizar una compra tardó 2 minutos,

ya que se paró a leer las diferentes páginas con resumen y finalización de la compra y por último visualizar la política de privacidad tardó 1 minuto.

8.3.1.1.2 Usuario2

- **¿Qué tipo de usuario está realizando las tareas?**

Usuario nivel alto que trabaja a diario con ordenador.

- **¿El usuario comienza a realizar las tareas de forma inmediata?**

Si, sin ningún problema.

- **¿El usuario ha cometido algún error al realizar alguna tarea? ¿Cuál?**

No ha cometido errores.

- **¿Cuánto tiempo ha tardado el usuario en realizar cada una de las tareas?**

El usuario ha realizado todas las tareas en tiempo inferior a 40 segundos, al estar acostumbrado a utilizar páginas del mismo estilo no ha tenido ningún problema.

8.3.1.1.3 Usuario3

- **¿Qué tipo de usuario está realizando las tareas?**

Usuario nivel medio que utiliza el ordenador y el móvil a menudo para ocio.

- **¿El usuario comienza a realizar las tareas de forma inmediata?**

Si, sin ningún problema.

- **¿El usuario ha cometido algún error al realizar alguna tarea? ¿Cuál?**

Al buscar la página con la política de privacidad ha pinchado varias veces en el menú principal.

- **¿Cuánto tiempo ha tardado el usuario en realizar cada una de las tareas?**

El usuario ha tardado 40 segundos en registrarse en la web, cerrar e iniciar sesión han sido 30 segundos, ver la descripción de un producto fueron 8 segundos, realizar una compra le llevó un total de 1 minuto y por último visualizar la política de privacidad tardó otro minuto.

8.3.1.1.4 Usuario4

- **¿Qué tipo de usuario está realizando las tareas?**

Usuario nivel medio que utiliza el ordenador y el móvil a menudo para ocio.

- **¿El usuario comienza a realizar las tareas de forma inmediata?**

Si, sin ningún problema.

- **¿El usuario ha cometido algún error al realizar alguna tarea? ¿Cuál?**

No

- **¿Cuánto tiempo ha tardado el usuario en realizar cada una de las tareas?**

El usuario ha tardado 52 segundos en registrarse en la web, cerrar e iniciar sesión han sido 35 segundos, ver la descripción de un producto fueron 9 segundos, realizar una compra le llevó un total de 1 minuto con 23 segundos y por último visualizar la política de privacidad tardó 15 segundos.

8.3.1.1.5 Usuario5

- **¿Qué tipo de usuario está realizando las tareas?**

Usuario nivel alto que trabaja a diario con ordenador.

- **¿El usuario comienza a realizar las tareas de forma inmediata?**

Si, sin ningún problema.

- **¿El usuario ha cometido algún error al realizar alguna tarea? ¿Cuál?**

No ha cometido errores.

- **¿Cuánto tiempo ha tardado el usuario en realizar cada una de las tareas?**

El usuario ha realizado todas las tareas en tiempo inferior a 48 segundos, haciendo todas las tareas sin dudar y dedicando el mayor tiempo a las tareas en las que se solicita introducir datos.

8.3.1.1.6 Conclusiones

Después de realizar todas las pruebas anteriores, se ha llegado a la conclusión de que el diseño del sitio es bueno, ya que ninguno de los usuarios ha estado perdido en ningún momento y se han tomado una serie de decisiones para mejorar algunos detalles:

1. Se ha añadido una nueva forma de pago, mediante Paypal, se permite a los usuarios utilizar su tarjeta de crédito para pagar las compras. Esta opción se había contemplado como una mejora, pero se ha creído necesaria su incorporación ya que cualquier tienda online dispone de varias opciones de pago.
2. Se va a dejar como mejora el envío de la factura por correo electrónico una vez finalizada la compra.
3. Debido a que los usuarios tardaban más tiempo en realizar las tareas que necesitan introducir texto en formularios, se han añadido un par de mejoras. La primera ha sido poner el foco automáticamente en el primer campo del formulario para que el usuario pueda empezar a escribir de inmediato, la segunda, aumentar un poco el tamaño de los campos del formulario para que el usuario pueda pinchar en ellos más fácil.

8.3.1.1.7 Guía Yusef Hassan Montero

Además de los cuestionarios diseñados anteriormente, se ha completado la siguiente guía de usabilidad desarrollada por Yusef Hassan Montero (7):

Criterios	¿Cumplido?
<u>Generales</u>	
¿Cuáles son los objetivos del sitio web? ¿Son concretos y bien definidos? ¿Los contenidos y servicios que ofrece se corresponden con esos objetivos?	Si
¿Tiene una URL correcta, clara y fácil de recordar? ¿Y las URL de sus páginas internas? ¿Son claras y permanentes?	No hay URL por ahora, las páginas internas si tienen URL claras y permanentes
¿Muestra de forma precisa y completa qué contenidos o servicios ofrece realmente el sitio web? El diseño de la página de inicio debe ser diferente al resto de páginas y cumplir la función de 'escaparate' del sitio.	Si, la página principal muestra las categorías de productos que se pueden encontrar
¿La estructura general del sitio web está orientada al usuario? Los sitios web deben estructurarse pensando en el usuario, sus objetivos y necesidades. La estructura interna de la empresa u organización, cómo funciona o se organiza no interesan al usuario.	Si
¿El <i>look & feel</i> general se corresponde con los objetivos, características, contenidos y servicios del sitio web? Ciertas combinaciones de colores ofrecen imágenes más o menos formales, serias o profesionales.	Si
¿Es coherente el diseño general del sitio web? Se debe mantener una coherencia y uniformidad en las estructuras y colores de todas las páginas. Esto sirve para que el usuario no se desoriente en su navegación.	Si
¿Es reconocible el diseño general del sitio web? Cuánto más se parezca el sitio web al resto de sitios web, más fácil será de usar.	Si
¿El sitio web se actualiza periódicamente? ¿Indica cuándo se actualiza? Las fechas que se muestren en la página deben corresponderse con actualizaciones, noticias, eventos...no con la fecha del sistema del usuario.	Si se actualiza, no se indica
<u>Identidad e Información</u>	
¿Se muestra claramente la identidad de la empresa-sitio a través de todas las páginas?	Si
El Logotipo, ¿es significativo, identificable y suficientemente visible?	No se ha diseñado un logo que sea muy identificable, pero si está visible en todas las páginas
El eslogan o <i>tagline</i> , ¿expresa realmente qué es la empresa y qué servicios ofrece?	No hay
¿Se ofrece algún enlace con información sobre la empresa, sitio web, 'webmaster',...?	Si

¿Se proporciona mecanismos para ponerse en contacto con la empresa? (email, teléfono, dirección postal, fax...)	Si
¿Se proporciona información sobre la protección de datos de carácter personal de los clientes o los derechos de autor de los contenidos del sitio web?	Si
En artículos, noticias, informes... ¿Se muestra claramente información sobre el autor, fuentes y fechas de creación y revisión del documento?	No hay
<i>Lenguaje y Redacción</i>	
¿El sitio web habla el mismo lenguaje que sus usuarios? Se debe evitar usar un lenguaje corporativista. Así mismo, hay que prestarle especial atención al idioma, y ofrecer versiones del sitio en diferentes idiomas cuando sea necesario.	Si
¿Emplea un lenguaje claro y conciso?	Si
¿Es amigable, familiar y cercano? Es decir, lo contrario a utilizar un lenguaje constantemente imperativo, mensajes crípticos, o tratar con "desprecio" al usuario.	Si
¿1 párrafo = 1 idea? Cada párrafo es un objeto informativo. Trasmite ideas, mensajes...Se deben evitar párrafos vacíos o varios mensajes en un mismo párrafo.	No aplica
<i>Rotulado</i>	
Los rótulos, ¿son significativos? Ejemplo: evitar rótulos del tipo "haga clic aquí".	Si
¿Usa rótulos estándar? Siempre que exista un "estándar" comúnmente aceptado para el caso concreto, como "Mapa del Sitio" o "Acerca de...".	Si
¿Usa un único sistema de organización, bien definido y claro? No se deben mezclar diferentes. Los sistemas de organización son: alfabético, geográfico, cronológico, temático, orientado a tareas, orientado al público y orientado a metáforas.	Alfabético
¿Utiliza un sistema de rotulado controlado y preciso? Por ejemplo, si un enlace tiene el rótulo "Quiénes somos", no puede dirigir a una página cuyo encabezamiento sea "Acerca de"	Si
El título de las páginas, ¿Es correcto? ¿Ha sido planificado? Relacionado con la capacidad para poder buscar y encontrar el sitio web.	Si
<i>Estructura y Navegación</i>	
La estructura de organización y navegación, ¿Es la más adecuada? Hay varios tipos de estructuras: jerárquicas, hipertextual, facetada,...	Si
En el caso de estructura jerárquica, ¿Mantiene un equilibrio entre Profundidad y Anchura?	Si
En el caso de ser puramente hipertextual, ¿Están todos los clúster de nodos comunicados? Aquí se mide la distancia entre nodos.	No aplica
¿Los enlaces son fácilmente reconocibles como tales? ¿Su caracterización indica su estado (visitados, activos,...)? Los enlaces no sólo deben reconocerse como tales, sino que su caracterización debe indicar su estado, y ser reconocidos como una unidad	Si, en el menú se añade la opción de activo para guiar al usuario

En menús de navegación, ¿Se ha controlado el número de elementos y de términos por elemento para no producir sobrecarga memorística? No se deben superar los 7±2 elementos, ni los 2 o, como mucho, 3 términos por elemento.	Si
¿Es predecible la respuesta del sistema antes de hacer clic sobre el enlace? Relacionado con el nivel de significación del rótulo del enlace, aunque también con: el uso de globos de texto, información contextual, la barra de estado del navegador,...	A veces
¿Se ha controlado que no haya enlaces que no lleven a ningún sitio? Enlaces que no llevan a ningún sitio: Los enlaces rotos, y los que enlazan con la misma página que se está visualizando (por ejemplo enlaces a la "home" desde la misma página de inicio)	Si
¿Existen elementos de navegación que orienten al usuario acerca de dónde está y cómo deshacer su navegación? ...como <i>breadcrumbs</i> , enlaces a la página de inicio,...recuerde que el logo también es recomendable que enlace con la página de inicio.	Si, breadcrumbs
Las imágenes enlace, ¿se reconocen como clicables? ¿Incluyen un atributo 'title' describiendo la página de destino? En este sentido, también hay que cuidar que no haya imágenes que parezcan enlaces y en realidad no lo sean.	No hay
¿Se ha evitado la redundancia de enlaces?	Si
¿Se ha controlado que no haya páginas "huérfanas"? Páginas huérfanas: que aún siendo enlazadas desde otras páginas, éstas no enlacen con ninguna.	No
Layout de la Página	
¿Se aprovechan las zonas de alta jerarquía informativa de la página para contenidos de mayor relevancia? (como por ejemplo la zona central)	Si
¿Se ha evitado la sobrecarga informativa? Esto se consigue haciendo un uso correcto de colores, efectos tipográficos y agrupaciones para discriminar información. Los grupos diferentes de objetos informativos de una página deben ser 7±2.	Si
¿Es una interfaz limpia, sin ruido visual?	Si
¿Existen zonas en "blanco" entre los objetos informativos de la página para poder descansar la vista?	Si
¿Se hace un uso correcto del espacio visual de la página? Es decir, que no se desaproveche demasiado espacio con elementos de decoración, o grandes zonas en "blanco", y que no se adjudique demasiado espacio a elementos de menor importancia.	Si
¿Se utiliza correctamente la jerarquía visual para expresar las relaciones del tipo "parte de" entre los elementos de la página? (La jerarquía visual se utiliza para orientar al usuario)	Si
¿Se ha controlado la longitud de página? Se debe evitar en la medida de lo posible el <i>scrolling</i> . Si la página es muy extensa, se debe fraccionar.	Si
Búsqueda (si es necesario, por la extensión del sitio, incorporar un buscador interno)	

¿Se encuentra fácilmente accesible? Es decir: directamente desde la home, y a ser posible desde todas las páginas del sitio, y colocado en la zona superior de la página.	Si
¿Es fácilmente reconocible como tal?	Si
¿Permite la búsqueda avanzada? (siempre y cuando, por las características del sitio web, fuera de utilidad que la ofreciera)	No
¿Muestra los resultados de la búsqueda de forma comprensible para el usuario?	Si
¿La caja de texto es lo suficientemente ancha?	Si
¿Asiste al usuario en caso de no poder ofrecer resultados para una consultada dada?	No
<i>Elementos Multimedia</i>	
¿Las fotografías están bien recortadas? ¿Son comprensibles? ¿Se ha cuidado su resolución?	Si
¿Las metáforas visuales son reconocibles y comprensibles por cualquier usuario? (prestar especial atención a usuarios de otros países y culturas)	Si
¿El uso de imágenes o animaciones proporciona algún tipo de valor añadido?	Si
¿Se ha evitado el uso de animaciones cíclicas?	No
<i>Ayuda</i>	
Si posee una sección de Ayuda, ¿Es verdaderamente necesaria? Siempre que se pueda prescindir de ella simplificando los elementos de navegación e interacción, debe omitirse esta sección.	No
En enlace a la sección de Ayuda, ¿Está colocado en una zona visible y "estándar"? La zona de la página más normal para incluir el enlace a la sección de Ayuda, es la superior derecha.	No hay
¿Se ofrece ayuda contextual en tareas complejas? (transferencias bancarias, formularios de registro...)	No hay
Si posee FAQs, ¿Es correcta tanto la elección como la redacción de las preguntas? ¿Y las respuestas?	No hay
<i>Accesibilidad (debería cubrirse con los test de Accesibilidad posteriores)</i>	
¿El tamaño de fuente se ha definido de forma relativa, o por lo menos, la fuente es lo suficientemente grande como para no dificultar la legibilidad del texto?	Si
¿El tipo de fuente, efectos tipográficos, ancho de línea y alineación empleadas facilitan la lectura?	Si
¿Existe un alto contraste entre el color de fuente y el fondo?	Si
¿Incluyen las imágenes atributos 'alt' que describan su contenido?	Si
¿Es compatible el sitio web con los diferentes navegadores? ¿Se visualiza correctamente con diferentes resoluciones de pantalla? Se debe prestar atención a: JScript, CSS, tablas, fuentes...	Si
¿Puede el usuario disfrutar de todos los contenidos del sitio web sin necesidad de tener que descargar e instalar <i>plugins</i> adicionales?	Si
¿Se ha controlado el peso de la página? Se deben optimizar las imágenes, controlar el tamaño del código JScript...	Si

¿Se puede imprimir la página sin problemas? Leer en pantalla es molesto, por lo que muchos usuarios preferirán imprimir las páginas para leerlas. Se debe asegurar que se puede imprimir la página (no salen partes cortadas), y que el resultado es legible.	Si
<u>Control y Retroalimentación</u>	
¿Tiene el usuario todo el control sobre el interfaz? Se debe evitar el uso de ventanas pop-up, ventanas que se abren a pantalla completa, banners intrusivos...	Si
¿Se informa constantemente al usuario acerca de lo que está pasando? Si el usuario tiene que esperar hasta que se termine una operación, se debe mostrar un mensaje indicándoselo y que debe esperar, con el tiempo de espera estimado o una barra de progreso.	Si
¿Se informa al usuario de lo que ha pasado? Por ejemplo, cuando un usuario valora un artículo o responde a una encuesta, se le debe informar de que su voto ha sido procesado correctamente.	Si
Cuando se produce un error, ¿se informa de forma clara y no alarmista al usuario de lo ocurrido y de cómo solucionar el problema? Siempre es mejor intentar evitar que se produzcan errores a tener que informar al usuario del error.	Si
¿Posee el usuario libertad para actuar? NO restringir la libertad del usuario: Uso de animaciones que no pueden ser "saltadas", páginas en las que desaparecen los botones de navegación, no impida al usuario poder usar el botón derecho de su ratón...	Si
¿Se ha controlado el tiempo de respuesta? Esto tiene que ver con el peso de cada página (accesibilidad) y tiene relación con el tiempo que tarda el servidor en finalizar una tarea y responder. El tiempo máximo que esperará un usuario son 10 segundos	Si
<u>Aclaraciones</u>	
¿Se ha evaluado adecuadamente la orientación del usuario? (Donde estoy, como volver, que he visitado, que va a pasar)	Si
¿Se ha usado correctamente la publicidad?	No hay

Tabla 50 Guía accesibilidad Yusef Hassan Montero

8.3.1.1.8 Conclusiones

La principal conclusión obtenida es que el diseño y desarrollo inicial se había hecho teniendo en cuenta la mayoría de las preguntas que aparecen en el test de Yusef Hassan Montero, por esta razón, hay muy pocos cambios que se han podido introducir tras su revisión.

Los cambios realizados son, en primer lugar, se ha añadido la fecha con la última actualización de la web y junto a ello, se ha añadido al generador un nuevo método encargado de poner automáticamente la fecha en la que se genera la parafarmacia.

Por otro lado, se ha estudiado la posibilidad de incluir una búsqueda avanzada, pero se ha descartado después de ver que otros sitios similares no disponen de ella, ya que no tiene mucho sentido por el tipo de productos que se venden, así que se ha decidido dejar las opciones de buscar por categorías o por nombre de producto.

8.3.1.1.9 Diseño adaptable

A continuación, podemos ver el resultado de pasar las diferentes herramientas para comprobar la adaptabilidad del sitio a diferentes dispositivos.

Prueba realizada sobre la página de productos de la parafarmacia web utilizando la herramienta de Google (8):

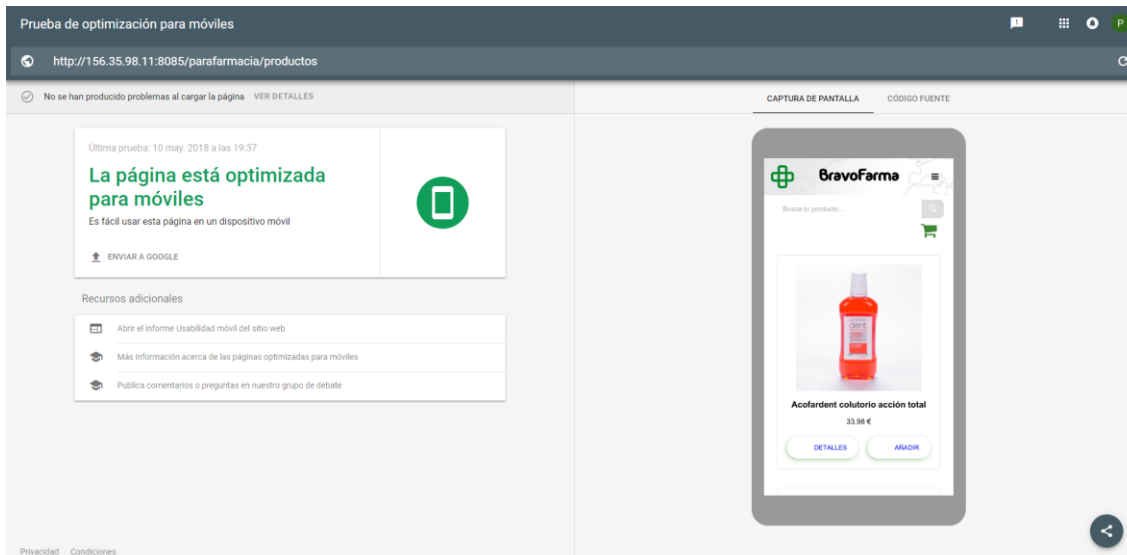


Ilustración 62 Resultado prueba de optimización para móviles

Prueba realizada sobre la herramienta Quirktools (9) para navegar por la aplicación desde las versiones de Tablet y móvil.

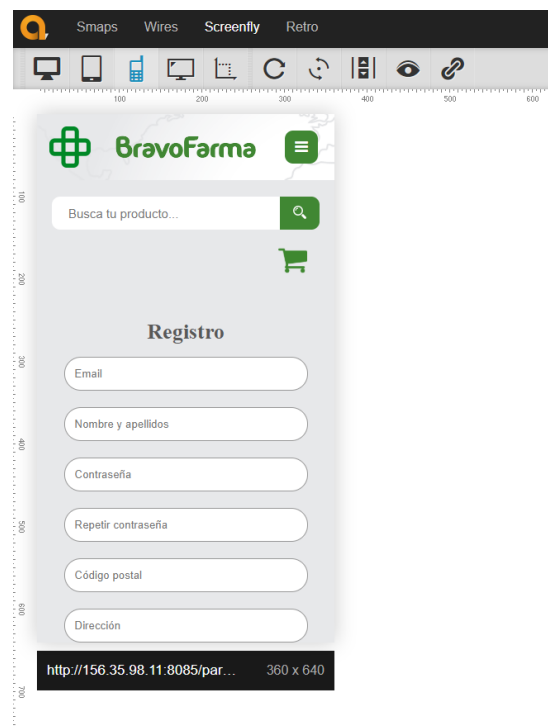


Ilustración 63 Prueba de optimización Quirktools

8.3.1.2 Cliente Android

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe dónde está dentro de la aplicación?</i>	Usuario1 Usuario2 Usuario3 Usuario4 Usuario5			
<i>¿Le resulta sencillo el uso de la aplicación?</i>	Usuario1 Usuario2 Usuario3 Usuario4 Usuario5			
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. Espera?</i>	Usuario1 Usuario2 Usuario3 Usuario4 Usuario5			
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>			Usuario5	Usuario1 Usuario2 Usuario3 Usuario4
Calidad del Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
<i>El tipo y tamaño de letra es</i>		Usuario2	Usuario1	

		Usuario3 Usuario4 Usuario5		
<i>Los iconos e imágenes usados son</i>	Usuario1 Usuario3 Usuario2 Usuario4 Usuario5			
<i>Los colores empleados son</i>	Usuario1 Usuario2 Usuario3 Usuario4 Usuario5			
Diseño de la Interfaz		Si	No	A veces
<i>¿Le resulta fácil de usar?</i>	Usuario1 Usuario2 Usuario3 Usuario4 Usuario5			
<i>¿El diseño de las pantallas es claro y atractivo?</i>	Usuario1 Usuario2 Usuario3 Usuario4			Usuario5
<i>¿Cree que el programa está bien estructurado?</i>	Usuario1 Usuario2 Usuario3			

	Usuario4		
	Usuario5		
Observaciones			
Un par de usuarios preguntaron por qué no se podía añadir la imagen del producto a la vez que lo creas.			
A un usuario le costaba un poco ver los textos de los pedidos.			

A continuación, se evalúan las siguientes tareas que se han pedido sobre los usuarios:

- *Ver los usuarios de la parafarmacia.*
- *Ver los pedidos de la parafarmacia.*
- Actualizar el primer pedido que encuentre al estado entregado.
- Ver los productos de la parafarmacia.
- Actualizar el nombre de uno de los productos.
- Añadir un nuevo producto.

8.3.1.2.1 Usuario1

- ***¿Qué tipo de usuario está realizando las tareas?***

Usuario medio-bajo que utiliza los dispositivos de vez en cuando para navegar.

- ***¿El usuario comienza a realizar las tareas de forma inmediata?***

Si.

- ***¿El usuario ha cometido algún error al realizar alguna tarea? ¿Cuál?***

No ha cometido errores.

- ***¿Cuánto tiempo ha tardado el usuario en realizar cada una de las tareas?***

Para las 4 primeras tareas, el usuario tardó un total de 40 segundos, ya que al no estar acostumbrado a utilizar aplicaciones tardaba un poco en volver atrás de una pantalla a otra. Para actualizar el nombre de un producto tardó 30 segundos ya que también tardó un poco en escribir y, por último, añadir un nuevo producto, le llevó 1 minuto y 30 segundos.

8.3.1.2.2 Usuario2

- **¿Qué tipo de usuario está realizando las tareas?**

Usuario nivel alto que trabaja a diario con ordenador.

- **¿El usuario comienza a realizar las tareas de forma inmediata?**

Si.

- **¿El usuario ha cometido algún error al realizar alguna tarea? ¿Cuál?**

No ha cometido errores.

- **¿Cuánto tiempo ha tardado el usuario en realizar cada una de las tareas?**

El usuario ha realizado todas las tareas en menos de 10 segundos a excepción de añadir un nuevo producto que ha tardado 20 segundos al tener que introducir datos del producto. Ha intentado primero añadirlo dejando los campos vacíos, pero no ha sido posible.

8.3.1.2.3 Usuario3

- **¿Qué tipo de usuario está realizando las tareas?**

Usuario nivel medio que utiliza el ordenador y el móvil a menudo para ocio.

- **¿El usuario comienza a realizar las tareas de forma inmediata?**

Si.

- **¿El usuario ha cometido algún error al realizar alguna tarea? ¿Cuál?**

No ha cometido errores.

- **¿Cuánto tiempo ha tardado el usuario en realizar cada una de las tareas?**

El usuario ha tardado una media de 10 segundos por actividad menos al actualizar y añadir producto que ha tardado 30 segundos.

8.3.1.2.4 Usuario4

- **¿Qué tipo de usuario está realizando las tareas?**

Usuario nivel medio que utiliza el ordenador y el móvil a menudo para ocio.

- **¿El usuario comienza a realizar las tareas de forma inmediata?**

Si, sin ningún problema.

- **¿El usuario ha cometido algún error al realizar alguna tarea? ¿Cuál?**

No.

- **¿Cuánto tiempo ha tardado el usuario en realizar cada una de las tareas?**

El usuario ha tardado 10 segundos en realizar todas las tareas menos actualizar producto que ha tardado 18 segundos y añadir producto que ha tardado 32 segundos.

8.3.1.2.5 Usuario5

- **¿Qué tipo de usuario está realizando las tareas?**

Usuario nivel alto que trabaja a diario con ordenador.

- **¿El usuario comienza a realizar las tareas de forma inmediata?**

Si, sin ningún problema.

- **¿El usuario ha cometido algún error al realizar alguna tarea? ¿Cuál?**

No ha cometido errores.

- **¿Cuánto tiempo ha tardado el usuario en realizar cada una de las tareas?**

El usuario ha realizado todas las tareas en menos de un minuto, la que más ha tardado ha sido añadir un producto, 20 segundos.

8.3.1.2.6 Conclusiones

Como conclusiones podemos decir que la aplicación es muy sencilla de utilizar para cualquier tipo de usuario, además, los usuarios para los que está dirigida deben tener ciertos conocimientos mínimos, por este motivo, no se va a realizar ningún cambio.

Con respecto a las observaciones realizadas, añadir que, el tamaño de los textos se puede configurar para todas las aplicaciones y ponerlo a mayor tamaño (cosa que en su dispositivo tenía hecho), además, se ha utilizado en todos los casos el tamaño recomendado en la guía de Material Design o superior. La opción de añadir las imágenes por separado se ha hecho así por ser la solución más eficiente encontrada a la hora de desarrollar esa parte y se ha ajustado de la mejor manera al diseño para facilitar al máximo su uso.

8.3.1.3 Generador

La prueba realizada para este apartado consiste en generar el código de una nueva parafarmacia. Se le pide que la nueva parafarmacia tenga tonos de colores verdes con un color de letra negro, un menú permanente que tenga 70px de altura, que se incluyan al menos dos categorías y un producto para cada una de ellas y que tenga los datos personales de la propia persona que está realizando la prueba.

Antes de realizar la prueba se facilita al usuario una breve guía de uso:

8.3.1.3.1 Guía generador

Para poder generar una nueva parafarmacia, se debe escribir un nuevo archivo con el lenguaje que se explicará a continuación. Este archivo no tendrá extensión “.txt” si no que será “.gen”. Para que la parafarmacia se genere, el archivo debe estar escrito sin ningún fallo y solamente con guardar su contenido, crearemos todo lo necesario. Si se produce algún fallo el programa nos avisará con una cruz roja en la línea del fallo indicando el error si nos situamos encima con el ratón.

Ahora, se procederá a describir cada una de las partes del lenguaje. Como estamos creando una parafarmacia, es lógico pensar que el código comience con una palabra clave “Parafarmacia” que recogerá el resto de partes que la forman. Dentro de la parafarmacia tendremos varias opciones a personalizar, el nombre de la tienda, el titular, el usuario principal, las categorías junto a los productos y la apariencia.

El esquema básico del lenguaje utilizado es el siguiente: La parafarmacia encierra a todas sus partes mediante llaves { ... }, cada una de las partes (apariencia, productos, categorías...) se encierra mediante los símbolos de mayor y menor < Apariencia ... >, las partes contienen los datos que podemos personalizar, estos datos irán separados por dos puntos < Apariencia = dato1 : dato2 >, por último, tenemos los datos complejos, que son aquellos formados por varios subdatos, por ejemplo, la apariencia de la parafarmacia se divide en colores, textos y márgenes, estos se consideran datos complejos ya que tenemos diferentes zonas con diferentes colores, color de fondo, color de productos, color de textos, color de menú... Los elementos complejos funcionan igual que los simples, pero se encierran en corchetes [Color = color1 : color2].

Por suerte, escribir el código para generar la parafarmacia es muy fácil, solo es necesario pulsar la combinación de teclas Ctrl + espacio y se desplegará un cuadro de diálogo con una ayuda para escribir el código. Esta ayuda contiene el esqueleto completo y solo será necesario escribir dentro de los recuadros que se mostrarán. Para facilitar más aun las cosas, para pasar de un dato a otro y empezar a escribir en el momento, solo tendremos que pulsar sobre la tecla del tabulador ↹.

Como información adicional, añadir que los colores se deben introducir con los códigos del color en hexadecimal: #FF5733.

8.3.1.3.2 Prueba

La prueba final se ha realizado sobre 3 de los usuarios utilizados en las pruebas anteriores: (Para facilitar la prueba se ha dado al usuario los códigos de colores que puede utilizar)

8.3.1.3.3 Usuario2

- ***¿El desarrollador comienza a escribir el código necesario de forma inmediata?***

Si.

- ***¿En qué partes se han cometido errores y cuáles?***

Al introducir el nombre de las categorías, ya que debe escribirse con la primera letra mayúscula como las palabras reservadas.

- ***¿Cuánto tiempo tarda el desarrollador en completar la tarea?***

El usuario ha tardado un total de 3 minutos 28 segundos

8.3.1.3.4 Usuario4

- ***¿El desarrollador comienza a escribir el código necesario de forma inmediata?***

Si.

- ***¿En qué partes se han cometido errores y cuáles?***

Al añadir un nuevo producto, mete el nuevo producto dentro de los mismos corchetes del primero: [Producto = ... Producto = ...] en lugar de [Producto = ...] [producto = ...]

También ha cometido error al introducir el nombre de las categorías.

- ***¿Cuánto tiempo tarda el desarrollador en completar la tarea?***

El usuario ha tardado un total de 3 minutos 28 segundos

8.3.1.3.5 Usuario5

- ***¿El desarrollador comienza a escribir el código necesario de forma inmediata?***

Si.

- ***¿En qué partes se han cometido errores y cuáles?***

Error al introducir el nombre de las categorías.

- ***¿Cuánto tiempo tarda el desarrollador en completar la tarea?***

El usuario tardó un total de 2 minutos 40 segundos.

8.3.1.3.6 Conclusiones

La conclusión que se obtiene tras realizar las pruebas con los usuarios es que el uso de las ayudas a la hora de escribir el código es la parte fundamental para evitar cometer errores. Se ha comprobado que todos los usuarios han cometido el mismo fallo a la hora de escribir las categorías porque no se ha explicado en la guía esa parte en concreto, por lo que se incluirá en la siguiente versión para facilitar más la escritura del código.

8.3.2 Pruebas de Accesibilidad

A continuación, se comenzará completando el cuestionario descrito en la sección de diseño de pruebas para comprobar que la parafarmacia web cumple con los aspectos más básicos de WCAG 2.0 nivel AA.

Cuestionario	
Pregunta	Respuesta
¿Los textos están escritos correctamente?	Si
¿Hay teclas de acceso rápido asignadas a los menús y a las funciones más utilizadas?	No las había, se ha corregido y se han añadido teclas de acceso rápido a los apartados del menú y se ha subrayado la letra con la que accedemos de forma rápida. Para las páginas con formularios se añade el foco al primer campo del formulario.
¿Cada elemento no textual tiene un equivalente en forma de texto?	Si, si deshabilitamos las imágenes del navegador, se sustituyen por el texto equivalente.
¿Evita el uso de medidas absolutas?	Si
¿Hay mecanismos de navegación consistentes?	Si
¿Se evita que las páginas se refresquen automáticamente?	Si
¿Evita que se abran pop-ups o se cambie de ventana sin informar al usuario?	Si
¿Usa un lenguaje simple y claro?	Si
¿Identifica claramente el destino de cada enlace?	Si

¿La información transmitida a través de colores está disponible sin color?	No hay información transmitida por colores
¿Hay suficiente contraste entre los colores de fondo y primer plano?	Si, todos cumplen con WCAG 2 AA probado con: https://snook.ca/technical/colour_contrast
¿Las páginas tienen un título corto y descripciones?	Los títulos de las páginas son lo suficiente descriptivos para este caso.
¿Las URIs de los sitios son lo más cortas posibles?	Si: /productos, /carrito, /inicio...
¿El contenido básico de la página se ve sin hacer scroll?	Si
¿Se evitan los enlaces externos?	Si
¿El tamaño global de la página es adecuado para las limitaciones de los dispositivos?	Si
¿Hay scroll en una única dirección?	Si
¿Se evitan las imágenes grandes o de alta resolución?	Salvo en la pantalla de inicio que pretendemos llamar la atención con una imagen grande.
¿Los formularios tienen valores por defecto?	Tienen placeholder para indicar al usuario que debe escribir.

Tabla 51 Resultado cuestionario accesibilidad

8.3.2.1 Revisión Preliminar

Navegación sin imágenes:

Como se puede comprobar en la siguiente imagen, al desactivar las imágenes, se sustituyen por un texto representativo:

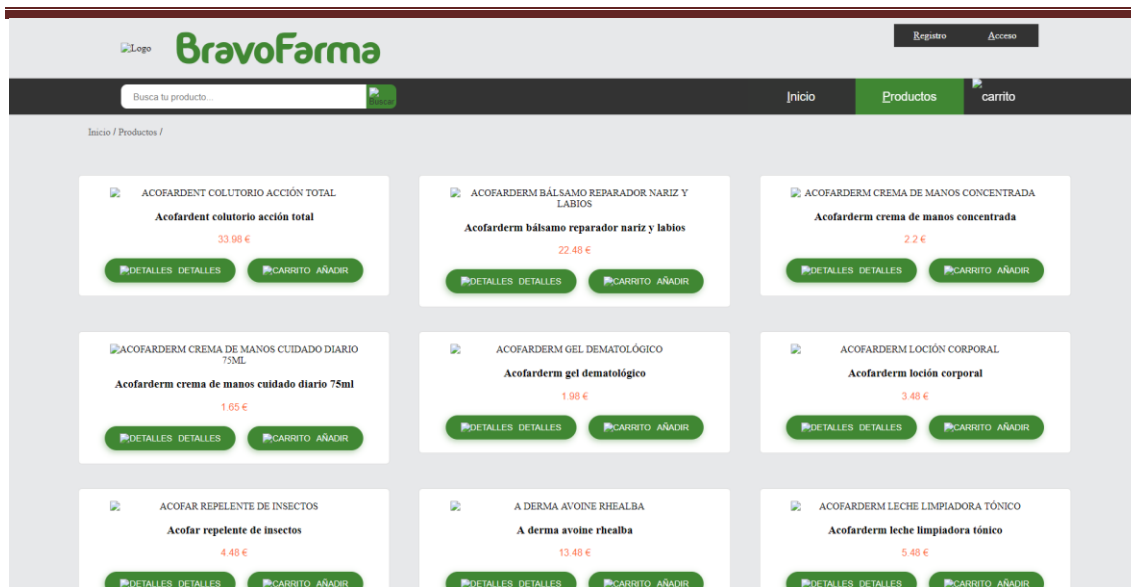


Ilustración 64 Parafarmacia web sin imágenes

Comprobación contrastes de colores:

Fondo verde con letra blanca mayor de 18pt cumple con AA:

https://snook.ca/technical/colour_contrast/colour.html#fg=FFFFFF,bg=408733

Fondo negro con letras blancas cumple con AAA:

https://snook.ca/technical/colour_contrast/colour.html#fg=FFFFFF,bg=333333

Fondo blanco con letras negras cumple con AAA:

https://snook.ca/technical/colour_contrast/colour.html#fg=333333,bg=FFFFFF

Cambiar el tamaño de la página

La siguiente imagen muestra la página de productos aumentada al 200%, se puede ver que sigue permitiendo la navegación sin problema y no aparece scroll horizontal:

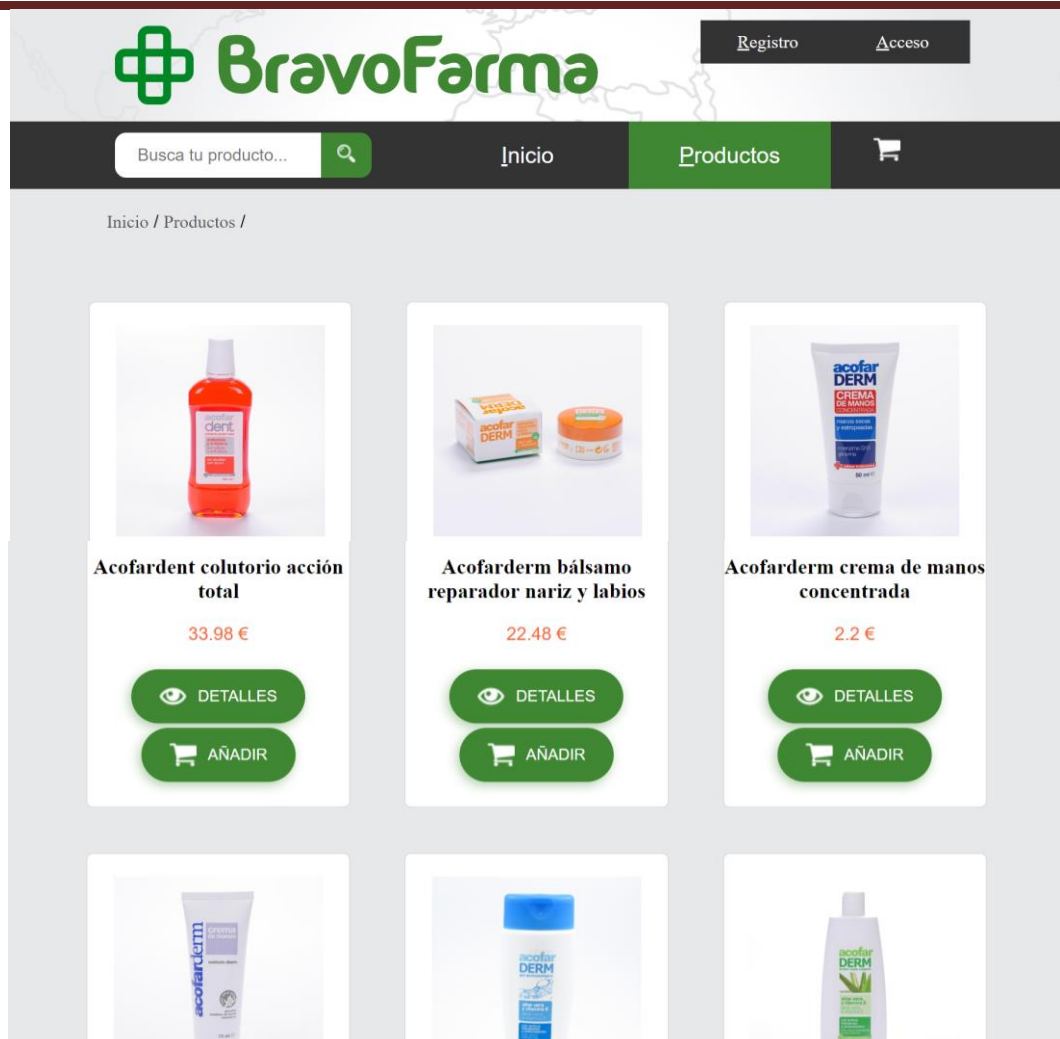


Ilustración 65 Parafarmacia web con el tamaño aumentado

Cambiar el tamaño de letra

En la siguiente imagen se ve la página de productos con el tamaño de letra ajustado desde el navegador a muy grande:

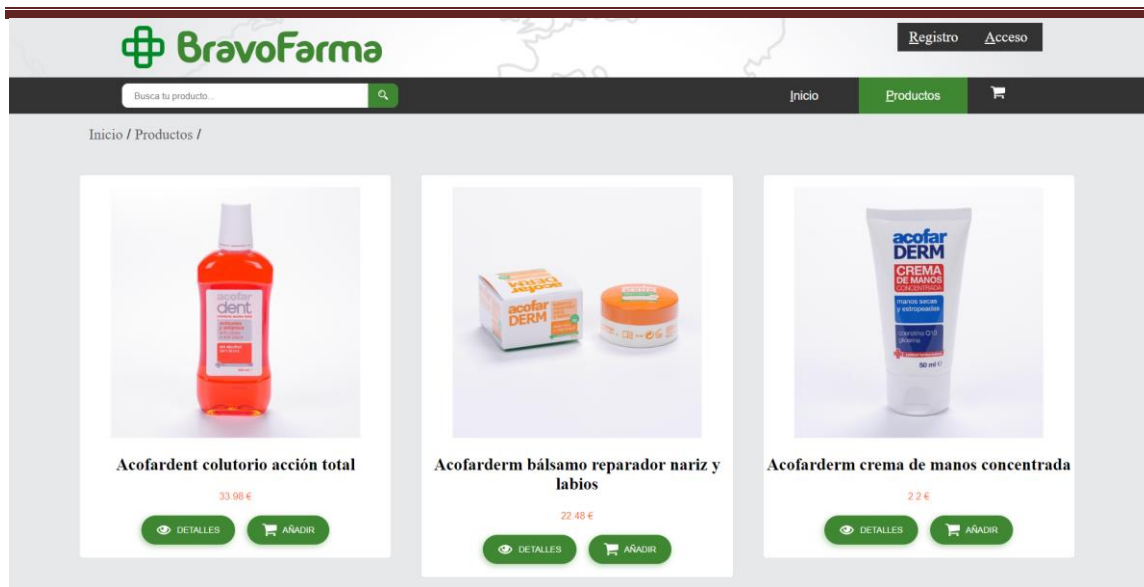


Ilustración 66 Parafarmacia web con el tamaño de letra aumentado

Cambiar el tamaño de la ventana

En las siguientes imágenes se ha modificado el tamaño de la ventana para ajustarlo al de una Tablet y un dispositivo móvil, se puede comprobar que se puede navegar sin ningún problema, los elementos no se descolocan si no que se ajustan a los nuevos tamaños.

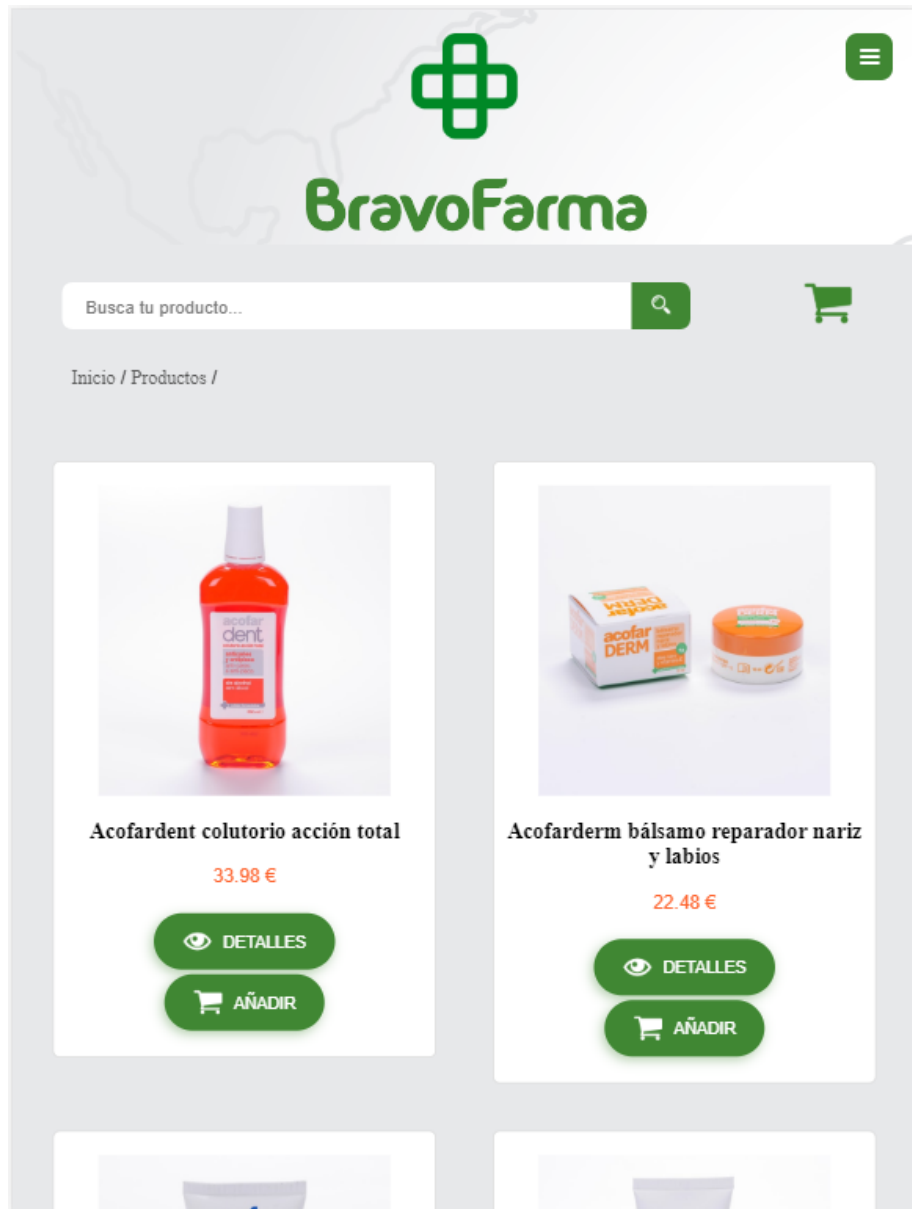


Ilustración 67 Parafarmacia web versión iPad

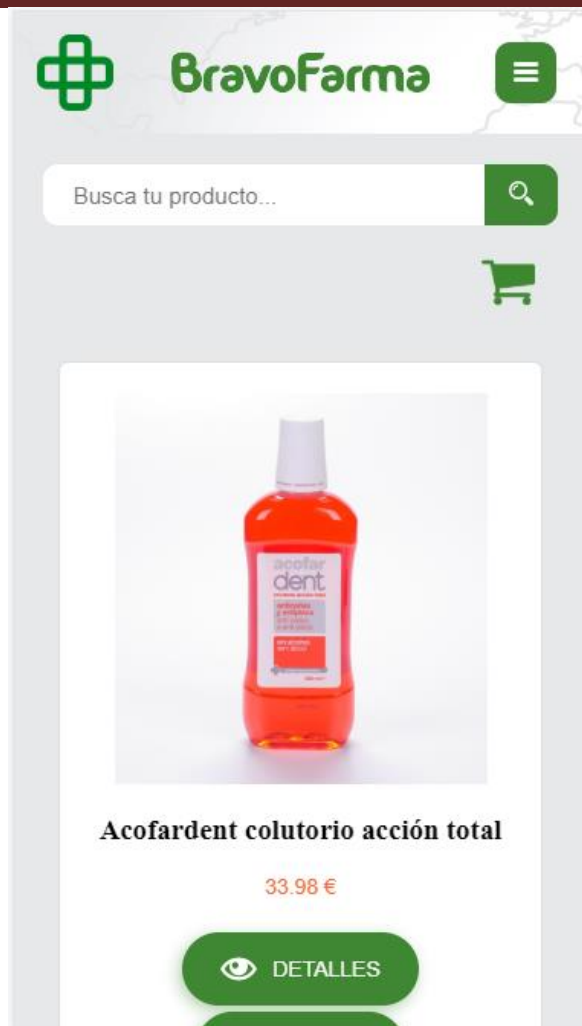
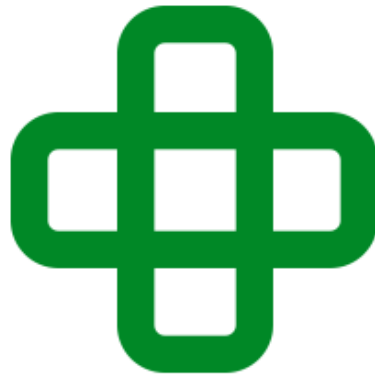


Ilustración 68 Parafarmacia web versión iPhone

Visualización sin hoja de estilos

A continuación, podemos ver la parafarmacia web sin hoja de estilos, el aspecto es completamente distinto, pero se puede navegar de igual manera



BravoFarma

- [Acceso](#)
- [Registro](#)
-
- [Productos](#)
 - [Salud](#)
 - [Cosmética](#)
 - [Higiene](#)
 - [Dietética](#)
 - [Bebes](#)
- [Inicio](#)
- [Inicio](#)
- [Productos](#)
- [Salud](#)
- [Cosmética](#)
- [Higiene](#)
- [Dietética](#)
- [Bebes](#)
- [Acceso](#)
- [Registro](#)

[Inicio](#) / [Productos](#) /



Acofardent colutorio acción total

33.98 €

[Detalles](#)

[Añadir](#)



Ilustración 69 Parafarmacia web sin hoja de estilos

Pruebas en diferentes navegadores

En las siguientes imágenes se puede ver las pantallas de los diferentes navegadores principales. En Internet Explorer observamos que no se pintan los colores ni se cogen los tipos de letra, esto se debe al uso de un nuevo elemento CSS, root, que permite crear variables con los diferentes valores que se reutilizarán durante toda la hoja de estilos como los colores. Aunque la solución sería muy sencilla, ya que se puede navegar perfectamente y que se pretende aprender y tratar de estar lo más actualizados posible, no se va hacer la versión personalizada de Internet Explorer.

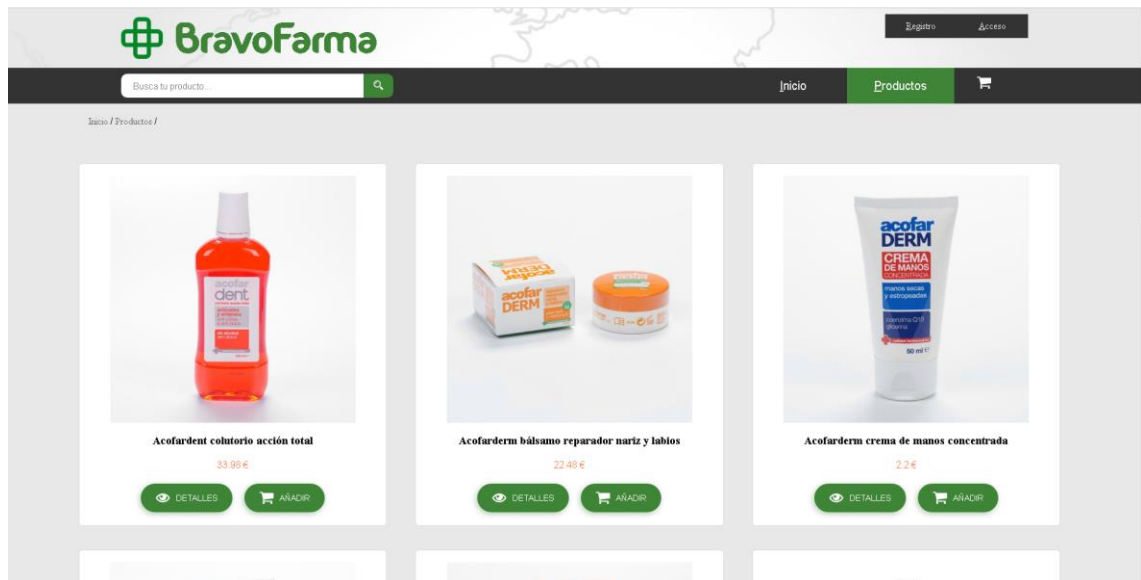


Ilustración 70 Parafarmacia web en Opera

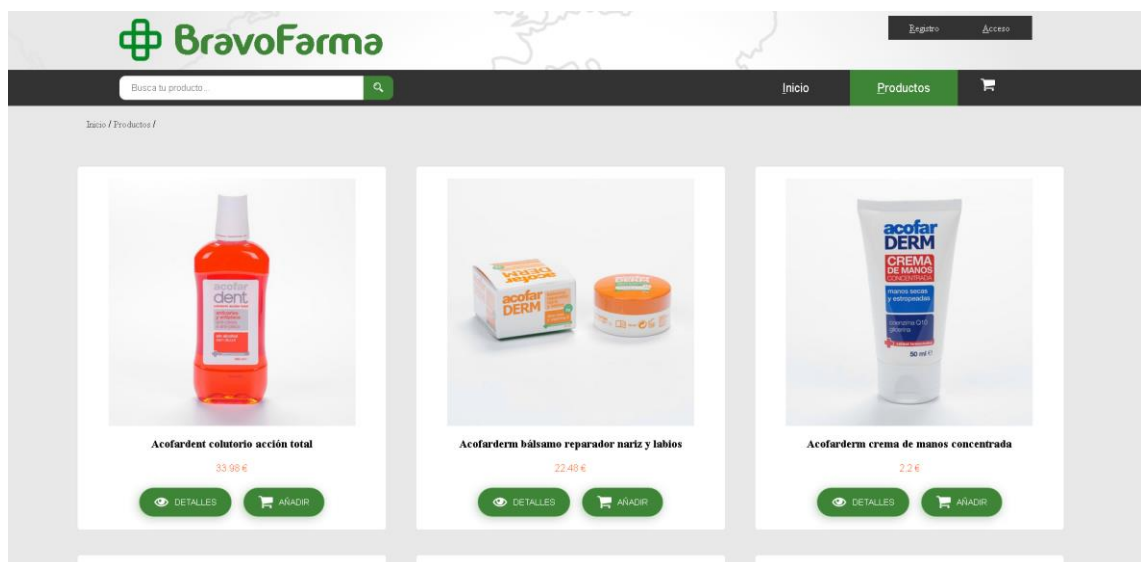


Ilustración 71 Parafarmacia web en Firefox

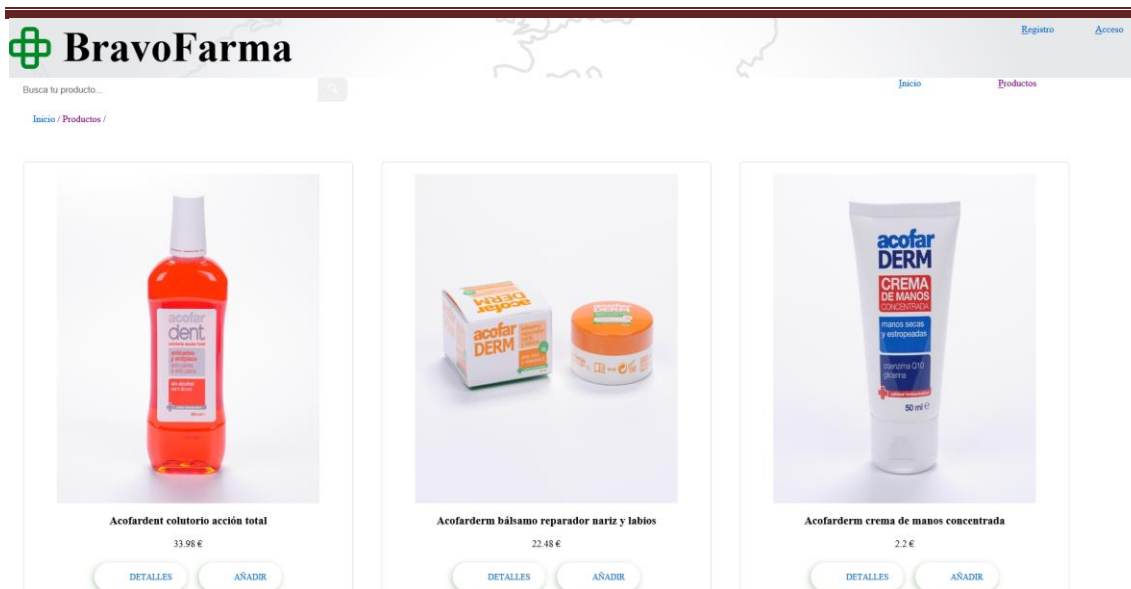


Ilustración 72 Parafarmacia web en Internet Explorer

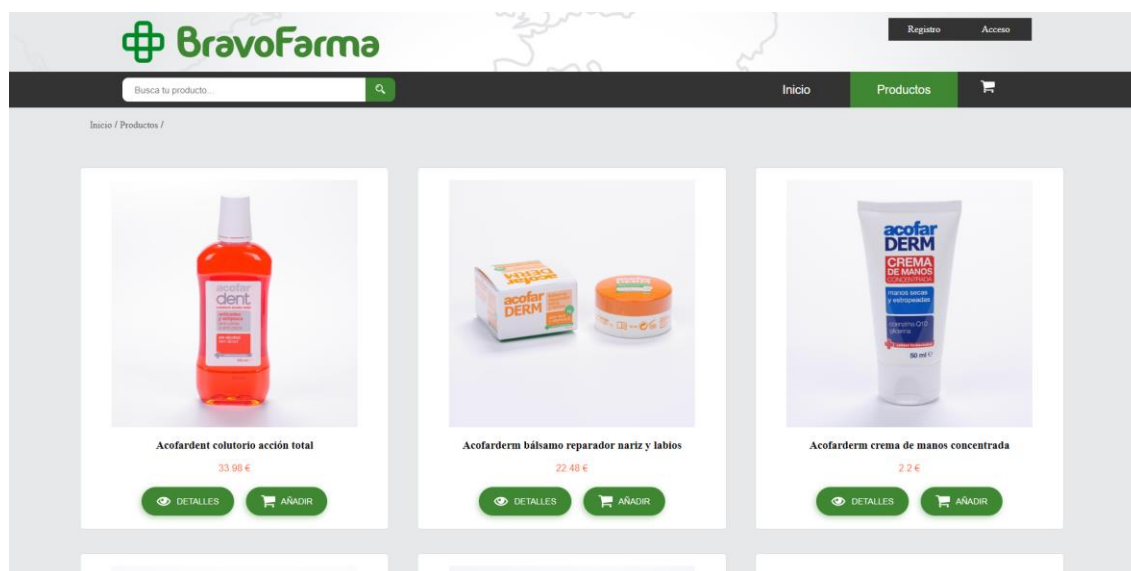


Ilustración 73 Parafarmacia web en Google Chrome

Herramienta WAVE

A continuación, se muestra el resultado de pasar el test de accesibilidad con la herramienta WAVE. Vemos que todas las páginas están limpias de errores, para ello ha sido necesario corregir algunos detalles en los formularios.



Ilustración 74 Parafarmacia web descripción examinada con WAVE

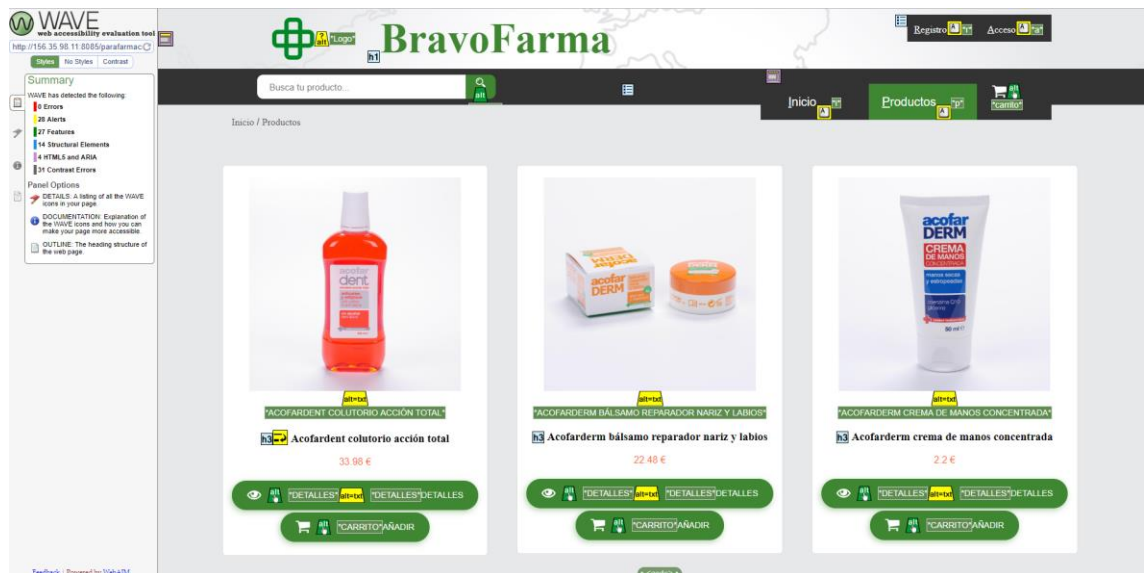


Ilustración 75 Parafarmacia web productos examinada con WAVE

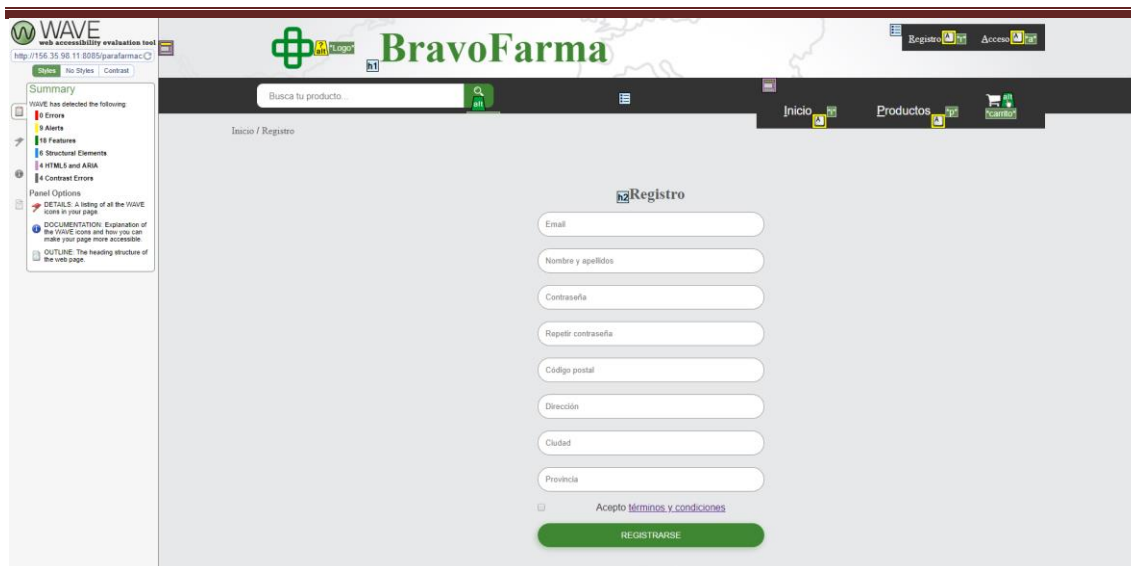


Ilustración 76 Parafarmacia web registro examinada con WAVE

Herramienta TAW

Se ha descartado esta herramienta ya que muestra problemas, pero no indica el lugar en el que aparecen y al revisarlos manualmente no se han podido encontrar.

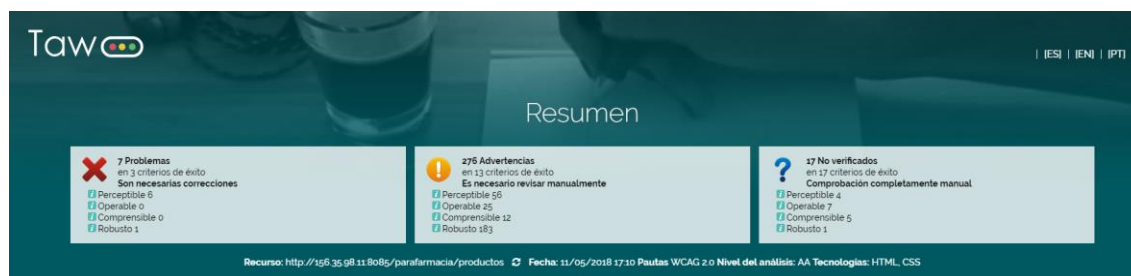


Ilustración 77 Parafarmacia web examinada con TAW

Herramienta HERA

Al pasar esta herramienta se han detectado 6 errores. Estos no se han podido solucionar, los cuatro primeros tienen que ver con la validación del CSS, y no es válido debido a que el validador utilizado no acepta todavía el uso de las variables con root.

Los otros dos errores no aparecen en la página, uno de ellos dice que el formulario con el buscador no tiene texto por defecto cuando si lo tiene y otro de ellos dice que hay enlaces juntos sin tener ningún texto entre ellos.

Summary of automatic analysis

<http://156.35.98.11:8085/parafarmacia/productos> Test again

HERA

During revision, bear in mind that scripts can generate dynamic content that needs the same considerations as static content does.

Summary

- URL: <http://156.35.98.11:8085/parafarmacia/productos>
- Date/Time: 11/05/2018 - 17:19 GMT
- Total: 208 elements
- Automatic analysis: 14 seconds
- Errors: 6 errors
- To check manually: 38 checkpoints
- Tester: (unknown)
- Navigator: Sin Identificar

Navigate by results

Use the links in the table to test each of the checkpoints manually or to check the results of automatic testing

Priority	Needs checking	Pass	Fail	N/A
P4 HERA_WS11.1	8 ρ	--	--	9 \checkmark
P2 HERA_WS11.1	19 ρ	1 \checkmark	4 \times	5 \checkmark
P3 HERA_WS11.1	11 ρ	2 \checkmark	2 \times	4 \checkmark

Navigate by guidelines

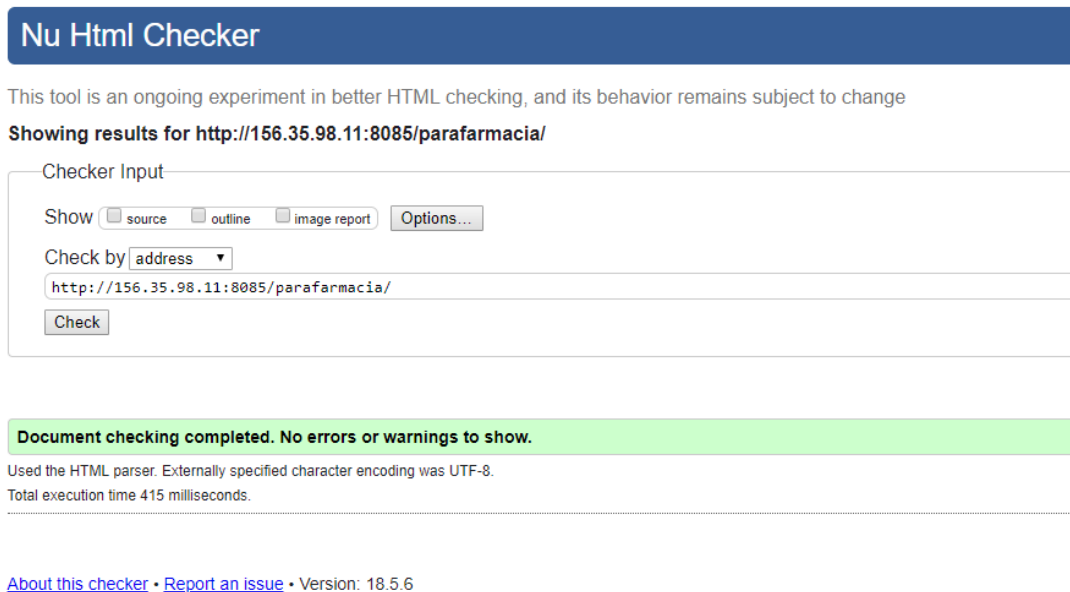
use the links to see each checkpoint. It shows all the checkpoints, independently of the results obtained in the automatic testing.

[Guidelines 1](#) [Guidelines 2](#) [Guidelines 3](#) [Guidelines 4](#) [Guidelines 5](#) [Guidelines 6](#) [Guidelines 7](#) [Guidelines 8](#) [Guidelines 9](#) [Guidelines 10](#) [Guidelines 11](#) [Guidelines 12](#) [Guidelines 13](#) [Guidelines 14](#)

Ilustración 78 Parafarmacia web examinada con HERA

Validación HTML

En las siguientes imágenes podemos ver dos páginas representativas de la parafarmacia web pasando el validador de HTML sin ningún problema.



Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <http://156.35.98.11:8085/parafarmacia/>

Checker Input

Show source outline image report

Check by

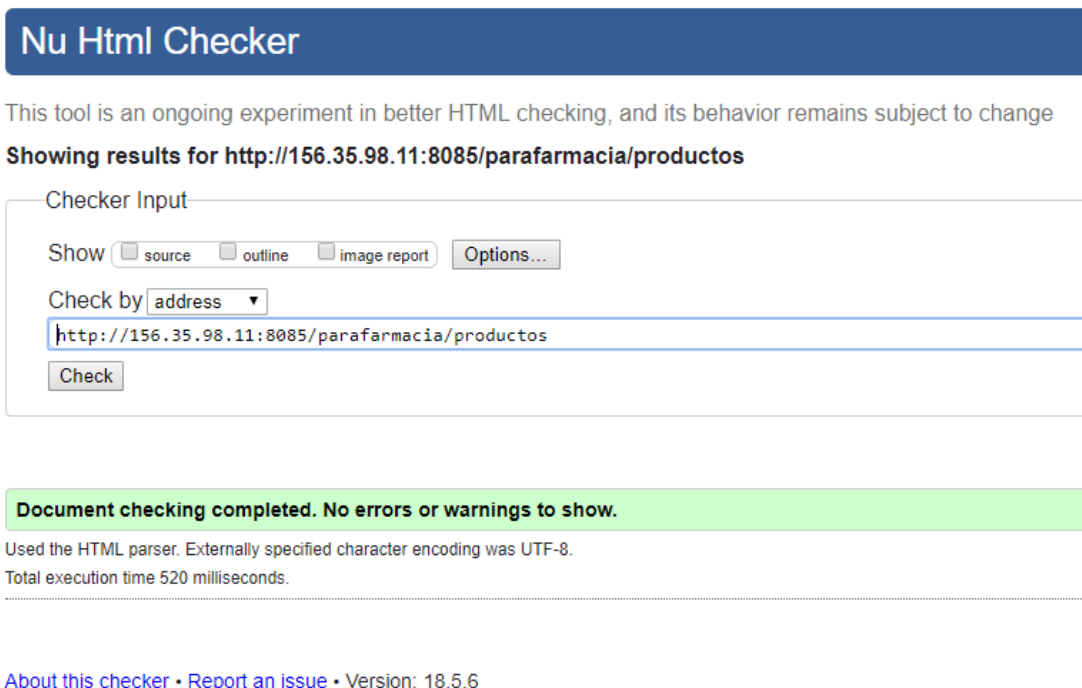
<http://156.35.98.11:8085/parafarmacia/>

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was UTF-8.
Total execution time 415 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 18.5.6

Ilustración 79 Parafarmacia web inicio validación HTML



Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <http://156.35.98.11:8085/parafarmacia/productos>

Checker Input

Show source outline image report

Check by

<http://156.35.98.11:8085/parafarmacia/productos>

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was UTF-8.
Total execution time 520 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 18.5.6

Ilustración 80 Parafarmacia web productos validación HTML

Validación CSS

Antes de pasar a validar el CSS, se muestra la siguiente tabla obtenida de la página <https://caniuse.com/#search=variable> en la que se muestra los navegadores que soportan el uso de variables. Como pudimos comprobar en los test anteriores, Internet Explorer no lo soporta, así como Opera Mini. Esta tabla se muestra debido a que el validador que pasamos a la página web no va a mostrar un resultado real, ya que los únicos errores que aparecen se deben a que no se encuentra actualizado para dar como válido las variables.

Browser	Version
IE	11
Edge	16, 17, 18
Firefox	59, 60, 61, 62
Chrome	49, 64, 65, 66, 67, 68, 69
Safari	11, 11.1, TP
iOS Safari	10.3, 11.2, 11.3
Opera Mini	all
Chrome for Android	66
UC Browser for Android	11.8
Samsung Internet	4, 6.2

Ilustración 81 Can I Use Variables CSS

En la siguiente imagen se muestran los errores, todos relacionados con el uso de variables. Vamos a utilizar dos validadores, <http://jigsaw.w3.org>, el primero no se encuentra actualizado pero el segundo si, <http://www.css-validator.org>.

El Servicio de Validación de CSS del W3C

Resultados del Validador CSS del W3C para <http://156.35.98.11:8085/parafarmacia/productos> (CSS versión 3)

Ir a: [Los Errores \(182\)](#) [Las Advertencias \(72\)](#) [Su Hoja de Estilo validada](#)

Resultados del Validador CSS del W3C para <http://156.35.98.11:8085/parafarmacia/productos> (CSS versión 3)

Disculpas! Hemos encontrado las siguientes errores (182)

URI	Linea	Columna	Detalle del error
4	.fe		La propiedad 'text-rendering' no existe :auto
46	body		Propiedad no válida : background-color Error de análisis sintáctico [var(--background-color)]
56	hr		Propiedad no válida : background-color Error de análisis sintáctico [var(--color-text)]
64	p		Propiedad no válida : font-family Error de análisis sintáctico [var(--familia-texto)]
79	nav		Propiedad no válida : background-color Error de análisis sintáctico [var(--nav-color)]
81	nav		Propiedad no válida : padding Error de análisis sintáctico [var(--margenes-laterales)]
82	nav		Propiedad no válida : height Error de análisis sintáctico [var(--altura-menu)]
90	nav a.elementoenu		Propiedad no válida : color Error de análisis sintáctico [var(--color-text-nav)]
91	nav a.elementoenu		Propiedad no válida : font-size Error de análisis sintáctico [var(--size-menu)]
92	nav a.elementoenu		Propiedad no válida : font-family Error de análisis sintáctico [var(--familia-texto)]
106	nav a.elementoenu:hover		Propiedad no válida : background-color Error de análisis sintáctico [var(--main-color)]
113	nav div.usuario		Propiedad no válida : margin Error de análisis sintáctico [var(--margenes-laterales) e e]
128	.subcabecera p a		Propiedad no válida : font-family Error de análisis sintáctico [var(--familia-titulos)]
129	.subcabecera p a		Propiedad no válida : color Error de análisis sintáctico [var(--color-titulo)]
158	.toggle		Propiedad no válida : background-color Error de análisis sintáctico [var(--main-color)]
159	.toggle		Propiedad no válida : color Error de análisis sintáctico [var(--color-text-nav)]
177	.menuov11		Propiedad no válida : background-color Error de análisis sintáctico [var(--nav-color)]
203	.menuov11 a		Propiedad no válida : color Error de análisis sintáctico [var(--color-text-nav)]
204	.menuov11 a		Propiedad no válida : font-family Error de análisis sintáctico [var(--familia-texto)]

Ilustración 82 Validador CSS Jigsaw



The W3C CSS Validation Service

W3C CSS Validator results for <http://156.35.98.11:8085/parafarmacia/productos> (CSS level 2.1)

W3C CSS Validator results for <http://156.35.98.11:8085/parafarmacia/productos> (CSS level 2.1)

Congratulations! No Error Found.

This document validates as [CSS level 2.1](#)!

To show your readers that you've taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the XHTML you could use to add this icon to your Web page:

```
<p>  
<a href="http://css-validator.org/check/referer">  
    
</a>  
</p>  
  
<p>  
<a href="http://css-validator.org/check/referer">  
    
</a>  
</p>
```

Ilustración 83 Validador CSS CSS-Validator

Puntos de verificación Prioridad 1:

Para acabar con las pruebas de accesibilidad, se va a completar un cuestionario para validar diferentes puntos de verificación en el cumplimiento del WCAG (10).

En general (Prioridad 1)	Sí	No	N/A
1.1 Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). <i>Esto incluye:</i> imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, <i>GIFs</i> animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos.	X		
2.1 Asegúrese de que toda la información transmitida a través de los colores también esté disponible sin color, por ejemplo, mediante el contexto o por marcadores.	X		
4.1 Identifique claramente los cambios en el idioma del texto del documento y en cualquier texto equivalente (por ejemplo, leyendas).			X
6.1 Organice el documento de forma que pueda ser leído sin hoja de estilo. Por ejemplo, cuando un documento HTML es interpretado sin asociarlo a una hoja de estilo, tiene que ser posible leerlo.	X		
6.2 Asegúrese de que los equivalentes de un contenido dinámico son actualizados cuando cambia el contenido dinámico.			X
7.1 Hasta que las aplicaciones de usuario permitan controlarlo, evite provocar destellos en la pantalla.	X		
14.1 Utilice el lenguaje apropiado más claro y simple para el contenido de un sitio.	X		
Y si utiliza imágenes y mapas de imagen (Prioridad 1)	Sí	No	N/A
1.2 Proporcione vínculos redundantes en formato texto para cada zona activa de un mapa de imagen del servidor.	X		
9.1 Proporcione mapas de imagen controlados por el cliente en lugar de por el servidor, excepto donde las zonas sensibles no puedan ser definidas con una forma geométrica.			X
Y si utiliza tablas (Prioridad 1)	Sí	No	N/A
5.1 En las tablas de datos, identifique los encabezamientos de fila y columna.	X		
5.2 Para las tablas de datos que tienen dos o más niveles lógicos de encabezamientos de fila o columna, utilice marcadores para asociar las celdas de encabezamiento y las celdas de datos.			X
Y si utiliza marcos ("frames") (Prioridad 1)	Sí	No	N/A
12.1 Titule cada marco para facilitar su identificación y navegación.			X
Y si utiliza "applets" y "scripts" (Prioridad 1)	Sí	No	N/A
6.3 Asegure que las páginas sigan siendo utilizables cuando se desconecten o no se soporten los scripts, <i>applets</i> u otros objetos programados. Si esto no es posible, proporcione información equivalente en una página alternativa accesible.	X		
Y si utiliza multimedia (Prioridad 1)	Sí	No	N/A
1.3 Hasta que las aplicaciones de usuario puedan leer en voz alta automáticamente el texto equivalente de la banda visual, proporcione una descripción auditiva de la información importante de la banda	X		

visual de una presentación multimedia.			
1.4 Para toda presentación multimedia dependiente del tiempo (por ejemplo, una película o animación) sincronice alternativas equivalentes (por ejemplo, subtítulos o descripciones de la banda visual) con la presentación.			X
Y si todo lo demás falla (Prioridad 1)	Sí	No	N/A
11.4 Si, después de los mayores esfuerzos, no puede crear una página accesible, proporcione un vínculo a una página alternativa que use tecnologías W3C, sea accesible, tenga información (o funcionalidad) equivalente y sea actualizada tan a menudo como la página (original) inaccesible.			X

Tabla 52 Puntos de verificación prioridad 1 WCAG

Puntos de verificación Prioridad 2:

En general (Prioridad 2)	Sí	No	N/A
2.2 Asegúrese de que las combinaciones de los colores de fondo y primer plano tengan el suficiente contraste para que sean percibidas por personas con deficiencias de percepción de color o en pantallas en blanco y negro [Prioridad 2 para las imágenes. Prioridad 3 para los textos].	X		
3.1 Cuando exista un marcador apropiado, use marcadores en vez de imágenes para transmitir la información.	X		
3.2 Cree documentos que estén validados por las gramáticas formales publicadas.	X		
3.3 Utilice hojas de estilo para controlar la maquetación y la presentación.	X		
3.4 Utilice unidades relativas en lugar de absolutas al especificar los valores en los atributos de los marcadores de lenguaje y en los valores de las propiedades de las hojas de estilo.	X		
3.5 Utilice elementos de encabezado para transmitir la estructura lógica y utilícelos de acuerdo con la especificación.	X		
3.6 Marque correctamente las listas y los ítems de las listas.	X		
3.7 Marque las citas. No utilice el marcador de citas para efectos de formato tales como sangrías.			X
6.5 Asegúrese de que los contenidos dinámicos son accesibles o proporcione una página o presentación alternativa.			X
7.2 Hasta que las aplicaciones de usuario permitan controlarlo, evite el parpadeo del contenido (por ejemplo, cambio de presentación en periodos regulares, así como el encendido y apagado).	X		
7.4 Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener las actualizaciones, no cree páginas que se actualicen automáticamente de forma periódica.	X		
7.5 Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener el redireccionamiento automático, no utilice marcadores para redirigir las páginas automáticamente. En su lugar, configure el servidor para que ejecute esta posibilidad.	X		
10.1 Hasta que las aplicaciones de usuario permitan desconectar la apertura de nuevas ventanas, no provoque apariciones repentinas de nuevas ventanas y no cambie la ventana actual sin informar al usuario.	X		
11.1 Utilice tecnologías W3C cuando estén disponibles y sean	X		

apropiadas para la tarea y use las últimas versiones que sean soportadas.			
11.2 Evite características desaconsejadas por las tecnologías W3C.	X		
12.3 Divida los bloques largos de información en grupos más manejables cuando sea natural y apropiado.	X		
13.1 Identifique claramente el objetivo de cada vínculo.	X		
13.2 Proporcione metadatos para añadir información semántica a las páginas y sitios.		X	
13.3 Proporcione información sobre la maquetación general de un sitio (por ejemplo, mapa del sitio o tabla de contenidos).		X	
13.4 Utilice los mecanismos de navegación de forma coherente.	X		
Y si utiliza tablas (Prioridad 2)	Sí	No	N/A
5.3 No utilice tablas para maquetar, a menos que la tabla tenga sentido cuando se alinee. Por otro lado, si la tabla no tiene sentido, proporcione una alternativa equivalente (la cual debe ser una versión alineada).	X		
5.4 Si se utiliza una tabla para maquetar, no utilice marcadores estructurales para realizar un efecto visual de formato.			X
Y si utiliza marcos ("frames") (Prioridad 2)	Sí	No	N/A
12.2 Describa el propósito de los marcos y cómo éstos se relacionan entre sí, si no resulta obvio solamente con el título del marco.			X
Y si utiliza formularios (Prioridad 2)	Sí	No	N/A
10.2 Hasta que las aplicaciones de usuario soporten explícitamente la asociación entre control de formulario y etiqueta, para todos los controles de formularios con etiquetas asociadas implícitamente, asegúrese de que la etiqueta está colocada adecuadamente.	X		
12.4 Asocie explícitamente las etiquetas con sus controles.	X		
Y si utiliza "applets" y "scripts" (Prioridad 2)	Sí	No	N/A
6.4 Para los <i>scripts</i> y <i>applets</i> , asegúrese de que los manejadores de eventos sean independientes del dispositivo de entrada.			X
7.3 Hasta que las aplicaciones de usuario permitan congelar el movimiento de los contenidos, evite los movimientos en las páginas.	X		
8.1 Haga los elementos de programación, tales como <i>scripts</i> y <i>applets</i> , directamente accesibles o compatibles con las ayudas técnicas [Prioridad 1 si la funcionalidad es importante y no se presenta en otro lugar; de otra manera, Prioridad 2].			X
9.2 Asegúrese de que cualquier elemento que tiene su propia interfaz pueda manejarse de forma independiente del dispositivo.			X
9.3 Para los "scripts", especifique manejadores de evento lógicos mejor que manejadores de eventos dependientes de dispositivos.			X

Tabla 53 Puntos de verificación prioridad 2 WCAG

Puntos de verificación Prioridad 3:

En general (Prioridad 3)	Sí	No	N/A
4.2 Especifique la expansión de cada abreviatura o acrónimo cuando aparezcan por primera vez en el documento.			X
4.3 Identifique el idioma principal de un documento.	X		
9.4 Cree un orden lógico para navegar con el tabulador a través de vínculos, controles de formulario y objetos.	X		

9.5 Proporcione atajos de teclado para los vínculos más importantes (incluidos los de los mapas de imagen de cliente), los controles de formulario y los grupos de controles de formulario.	X		
10.5 Hasta que las aplicaciones de usuario (incluidas las ayudas técnicas) interpreten claramente los vínculos contiguos, incluya caracteres imprimibles (rodeados de espacios), que no sirvan como vínculo, entre los vínculos contiguos.		X	
11.3 Proporcione la información de modo que los usuarios puedan recibir los documentos según sus preferencias (por ejemplo, idioma, tipo de contenido, etc.).			X
13.5 Proporcione barras de navegación para destacar y dar acceso al mecanismo de navegación.	X		
13.6 Agrupe los vínculos relacionados, identifique el grupo (para las aplicaciones de usuario) y, hasta que las aplicaciones de usuario lo hagan, proporcione una manera de evitar el grupo.	X		
13.7 Si proporciona funciones de búsqueda, permita diferentes tipos de búsquedas para diversos niveles de habilidad y preferencias.		X	
13.8 Localice la información destacada al principio de los encabezamientos, párrafos, listas, etc.	X		
13.9 Proporcione información sobre las colecciones de documentos (por ejemplo, los documentos que comprendan múltiples páginas).			X
13.10 Proporcione un medio para saltar sobre un <i>ASCII art</i> de varias líneas.			X
14.2 Complemente el texto con presentaciones gráficas o auditivas cuando ello facilite la comprensión de la página.	X		
14.3 Cree un estilo de presentación que sea coherente para todas las páginas.	X		
Y si utiliza imágenes o mapas de imagen (Prioridad 3)	Sí	No	N/A
1.5 Hasta que las aplicaciones de usuario interpreten el texto equivalente para los vínculos de los mapas de imagen de cliente, proporcione vínculos de texto redundantes para cada zona activa del mapa de imagen de cliente.			X
Y si utiliza tablas (Prioridad 3)	Sí	No	N/A
5.5 Proporcione resúmenes de las tablas.			X
5.6 Proporcione abreviaturas para las etiquetas de encabezamiento.			X
10.3 Hasta que las aplicaciones de usuario (incluidas las ayudas técnicas) interpreten correctamente los textos contiguos, proporcione un texto lineal alternativo (en la página actual o en alguna otra) para <i>todas</i> las tablas que maquetan texto en paralelo, en columnas de palabras.			X
Y si utiliza formularios (Prioridad 3)	Sí	No	N/A
10.4 Hasta que las aplicaciones de usuario manejen correctamente los controles vacíos, incluya caracteres por defecto en los cuadros de edición y áreas de texto.	X		

Tabla 54 Puntos de verificación prioridad 3 WCAG

Conclusiones

Para que una página web tenga un nivel de accesibilidad, tiene que cumplir con todos los puntos de verificación anteriores, como queremos que la parafarmacia tenga un nivel AA, deberemos cumplir con todos los puntos de prioridad 1 y 2. Como se puede ver, la web cumple

con todos los apartados aplicables de prioridad 1, pero hay alguno de prioridad 2 que no. Estos apartados que no se cumplen son, la creación del mapa del sitio, cosa que incluiremos antes de finalizar el proyecto y proporcionar metadatos para añadir información semántica, se incluirá un JSON con dicha información antes de finalizar el proyecto. Una vez cumplidos con ambos apartados, podremos concluir que la web tendrá el nivel de accesibilidad esperado.

8.4 Pruebas de Rendimiento

Antes de proceder a realizar las pruebas manualmente para medir los diferentes tiempos de carga, se han pasado dos herramientas para calcular el rendimiento, Pagespeed y Pingdom. El resultado de estas pruebas depende en gran medida de la máquina utilizada como servidor para alojar la aplicación. Esta máquina es un ordenador DELL con Windows 7 de 64 bits, 8 GB de RAM, 500 GB de memoria y un procesador Intel Core i7. Las características de un servidor real son superiores a las del equipo para pruebas, por lo que se espera que un resultado positivo sobre él, será un resultado positivo sobre cualquier servidor en el que alojemos finalmente la parafarmacia.

8.4.1 PageSpeed

Como se puede ver en la siguiente imagen, el resultado de la prueba es muy positivo, ya que cuanto más cerca de 100, mejor optimizado estará el sitio. Además, sugiere unas cuantas mejoras: usar el almacenamiento en caché, comprimir los recursos, optimizar las imágenes (ya realizado), minificar el CSS y el HTML. Estas mejoras se tendrán en cuenta en futuras actualizaciones de la parafarmacia.

The screenshot shows the PageSpeed Insights interface. At the top, there's a navigation bar with 'PÁGINA PRINCIPAL', 'GUÍAS', 'REFERENCIA', and 'ASISTENCIA'. Below that, the URL 'http://156.35.98.11:8085/parafarmacia/productos' is entered in a search bar, with an 'ANALIZAR' button. The device selection is set to 'Ordenador'. The performance metrics are displayed as follows:

Métrica	Estado
Velocidad	Unavailable
Optimización	Good (83 / 100)

Below the metrics, there is a note: 'Los datos sobre el rendimiento real de esta página no estaban disponibles. No obstante, PageSpeed Insights ha podido analizarla para encontrar posibles optimizaciones. Si se aplican, puede mejorar la velocidad de la página. Consulta las recomendaciones a continuación. Más información'. To the right, there is a small image of a laptop displaying the website being tested.

Estadísticas de la página

PSI calcula que esta página necesita 1 ciclos de ida y vuelta por el bloqueo del renderizado y aproximadamente 25 recursos (0,3 MB) para cargarse. La página mediana necesita 4 ciclos de ida y vuelta y aproximadamente 89 recursos (1,3 MB). Cuantos menos ciclos de ida y vuelta y bytes necesite una página, más rápida será.

Sugerencias de optimización

- Aprovechar el almacenamiento en caché del navegador
▶ [Mostrar cómo corregirlo](#)
- Habilitar compresión
▶ [Mostrar cómo corregirlo](#)
- Eliminar el JavaScript que bloquea la visualización y el CSS del contenido de la mitad superior de la página
▶ [Mostrar cómo corregirlo](#)

Ilustración 84 Test de rendimiento PageSpeed

8.4.2 PingDom

Como podemos observar en la siguiente imagen, el tiempo medio de carga de las páginas es de 1.68 segundos, inferior al tiempo esperado de 2 segundos, además, podemos ver varias mejoras que se pueden realizar, almacenamiento en caché, quitar redireccionamientos a otras páginas (el redireccionamiento que se ha incluido es a los test de validación HTML y CSS) y el uso de cookies

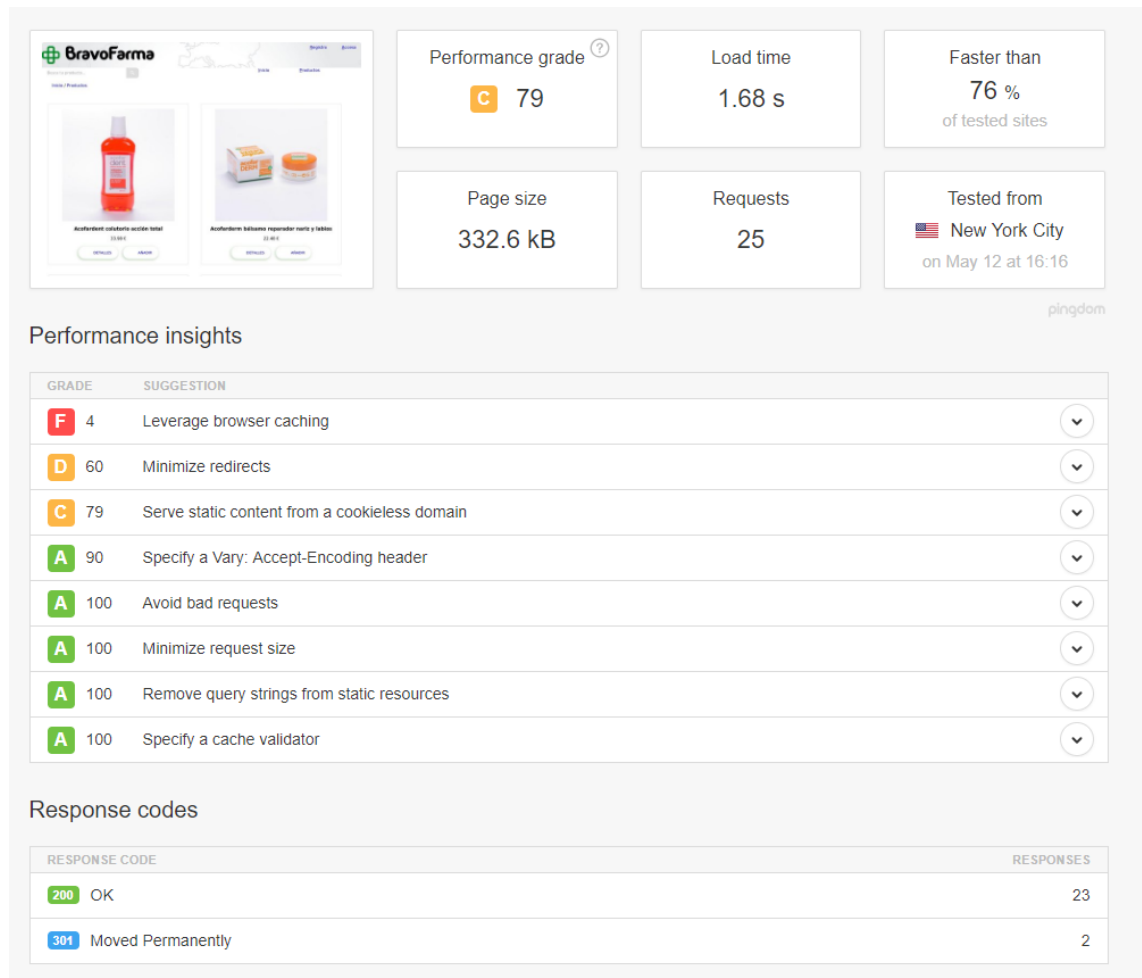


Ilustración 85 Test de rendimiento PingDom

8.4.3 Formulario

El siguiente formulario de tiempos se realizará mediante pruebas manuales desde tres dispositivos diferentes, para obtener el resultado se hará la media del tiempo de carga entre los tres.

Cuestionario	
Medir	Tiempo
Tiempo de carga de los productos en la parafarmacia web.	0.76 segundos
Tiempo de carga de los productos en el cliente Android.	La pantalla tarda en mostrarse 1 segundo, pero la carga de las imágenes de los productos se carga mediante diferentes solicitudes, la primera tarda 2 segundos. La segunda vez que accedemos las imágenes tardan en cargarse 1.2 segundos ya que se encuentran cacheadas.
Tiempo de carga de los pedidos en el cliente Android.	0.9 segundos
Tiempo que tarda un producto en insertarse en la base de datos desde el cliente Android hasta la parafarmacia web.	2.4 segundos, aunque este tiempo no es el tiempo de espera para el usuario ya que se realiza en una tarea asíncrona que no para la aplicación.
Tiempo que tarda el cliente Android en copiar los datos de la parafarmacia web a su base de datos.	3 segundos de media, aunque se deja 5 segundos la pantalla principal con el logo para que tenga tiempo de sobra.
Tiempo que tarda el generador en crear una nueva parafarmacia.	Una media de 7 segundos ya que el entorno de desarrollo debe actualizarse y compilar el código. El tiempo que tarda en generar los archivos es mucho menor, 0.5 segundos
Tiempo que tarda la parafarmacia web en registrar un nuevo usuario.	0.8 segundos

Tabla 55 Resultado pruebas de rendimiento

8.4.3.1 Conclusiones

Podemos concluir, una vez obtenidos las mediciones de tiempo, que la aplicación no tiene ningún punto con sobrecargas que provoque tiempos de carga superiores a los esperados, por lo que no se realizará ninguna acción como consecuencia de estas pruebas.

Capítulo 9. Manuales del Sistema

9.1 Manual de Instalación y ejecución

A continuación, vamos a ver los procesos a seguir para poder instalar los diferentes subsistemas que componen el proyecto.

9.1.1 Parafarmacia Web

La parafarmacia web ha sido desarrollada con el framework Spring y el lenguaje de programación JAVA. Tenemos dos alternativas distintas para poder instalar el sistema. La primera, más sencilla, es disponer de un Host o un servidor que permita ejecutar aplicaciones JAVA, de esta forma, solo sería necesario guardar en el servidor el archivo `.war` que contiene los archivos que forman la parafarmacia y crear la base de datos MySQL importando el archivo `.sql` con las tablas.

La segunda forma, para instalarlo en local, comienza con la instalación del entorno de desarrollo Spring Tool Suite.

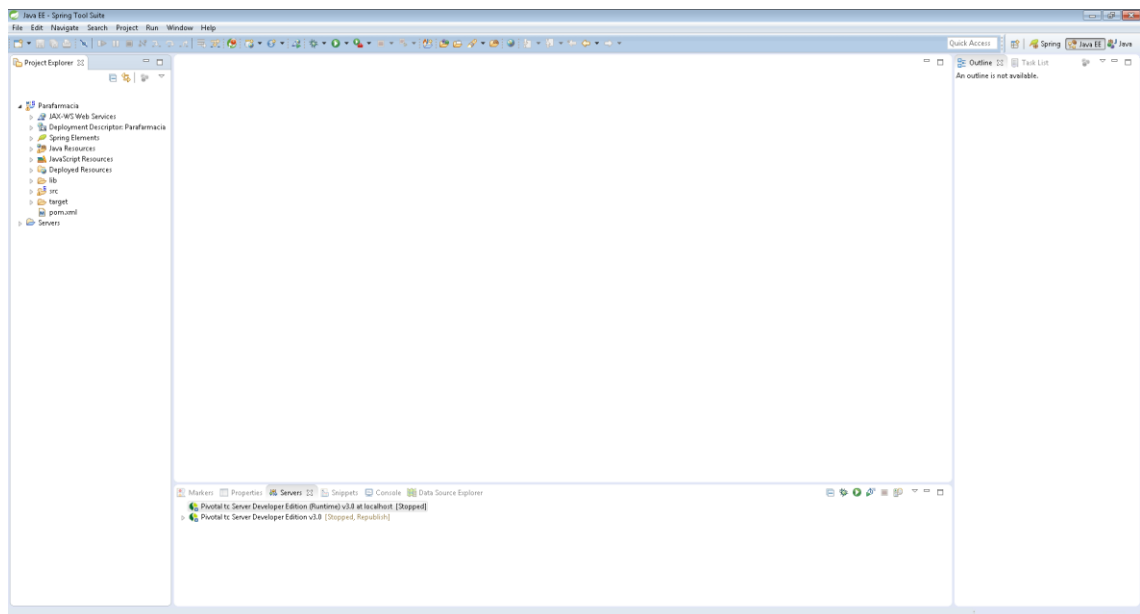


Ilustración 86 Entorno de desarrollo Spring Tool Suite

A continuación, debemos importar el proyecto con la parafarmacia web y el proyecto Servers que contiene un servidor Tomcat que permite ejecutar en local el proyecto. Cuando tengamos ambos proyectos importados, pinchamos con el segundo botón encima de la parafarmacia y seleccionamos *Run As -> Run on Server*.

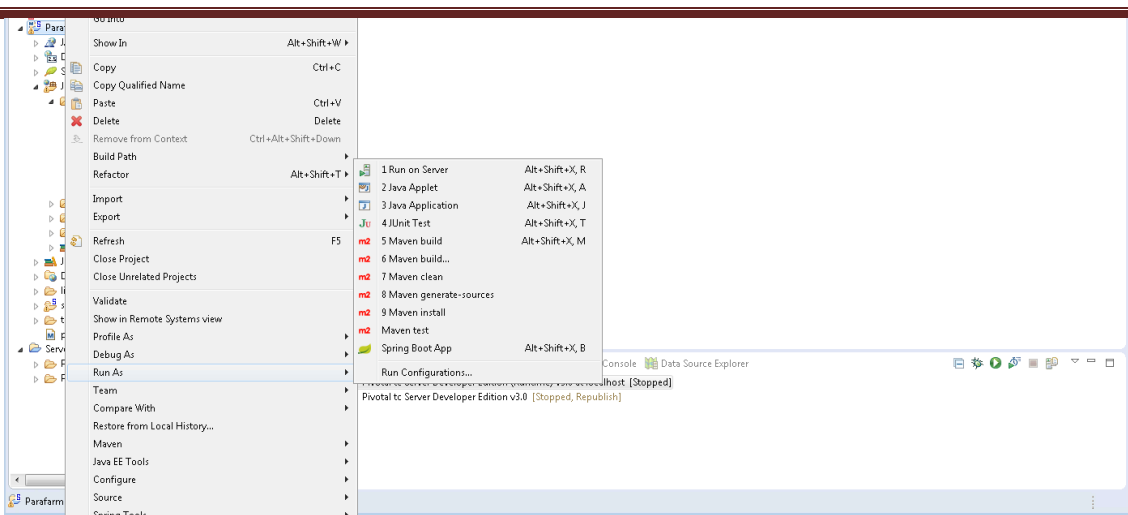


Ilustración 87 Entorno de desarrollo Spring Tool Suite Run As...

Con esto, se ejecutará la aplicación y nos aparecerá una ventana con la parafarmacia web funcionando casi del todo.

El penúltimo paso que debemos hacer es instalar una base de datos MySQL e importar el archivo `.sql` que contiene el script de generación de las tablas, en mi caso, he utilizado la base de datos MySQL que ofrece XAMPP.

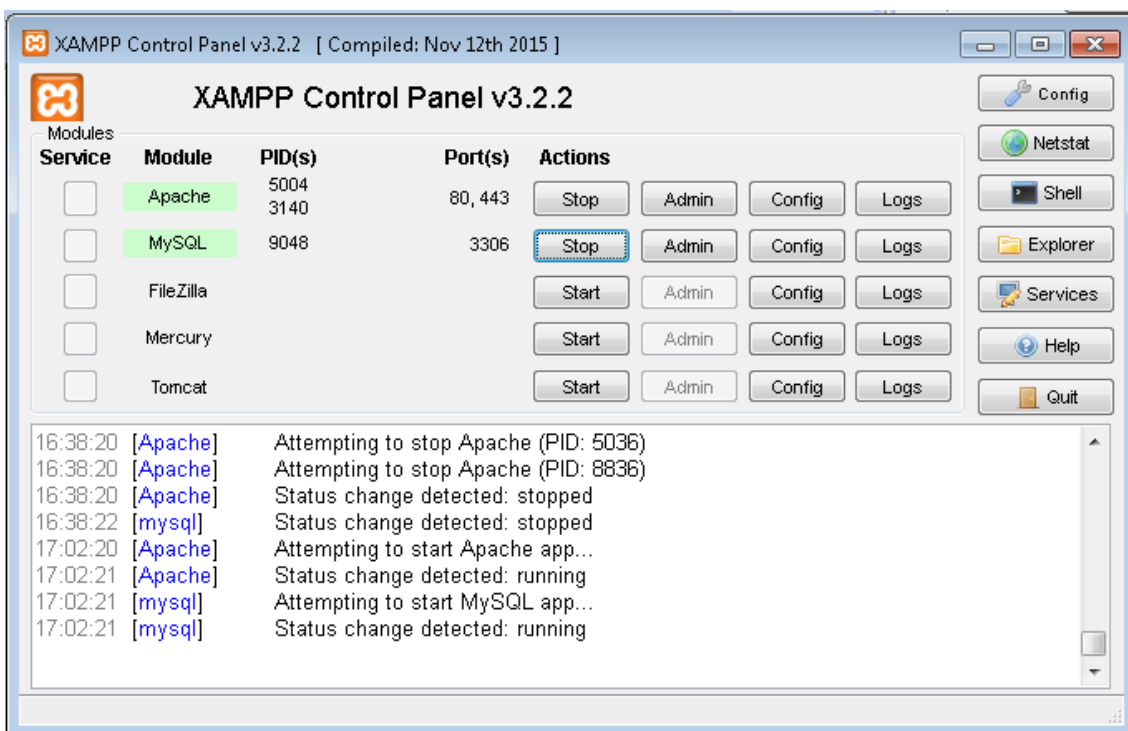


Ilustración 88 XAMPP con MySQL inicializado

Para finalizar, es necesario modificar el archivo `upload.php` encargado de subir imágenes a la parafarmacia para indicar la ruta en la que se almacenarán. Dicha ruta tiene una parte fija que es la que apunta a la carpeta de imágenes de los productos:

`Parafarmacia/src/main/webapp/resources/product/`

Este archivo se podrá desplegar en la raíz del servidor y será utilizado por el cliente Android para subir las imágenes de los productos.

```
<?php
$target_dir = "./Parafarmacia/src/main/webapp/resources/product/";
```

Ilustración 89 Ruta de almacenamiento de imágenes

9.1.2 Cliente Android

La instalación del cliente Android es muy sencilla.

Tenemos un archivo `parafarmacia.apk`, dicho archivo debemos copiarlo a nuestro dispositivo Android y abrirlo en él.

Una vez abierto, nos mostrará un mensaje indicando que se va a instalar una nueva aplicación y le daremos a continuar.

Cuando la aplicación esté instalada nos aparecerá en el menú de nuestro dispositivo junto al resto de aplicaciones y solo tendremos que pulsar sobre ella para abrirla.



 output.json	18/04/2018 17:34	JSON File	1 KB
 parafarmacia.apk	18/04/2018 17:34	Archivo APK	3.135 KB

Ilustración 90 Archivo .apk para instalar nuestro cliente Android

9.1.3 Generador

Para instalar el generador de parafarmacias, en primer lugar, es necesario tener Eclipse instalado en nuestro equipo. Se puede descargar en la página oficial <https://eclipse.org>.

Una vez instalado, lo abrimos y nos vamos a *Help -> Eclipse Marketplace*, ahí se nos abrirá una ventana como en la siguiente imagen, debemos buscar el plugin de Xtext e instalarlo:

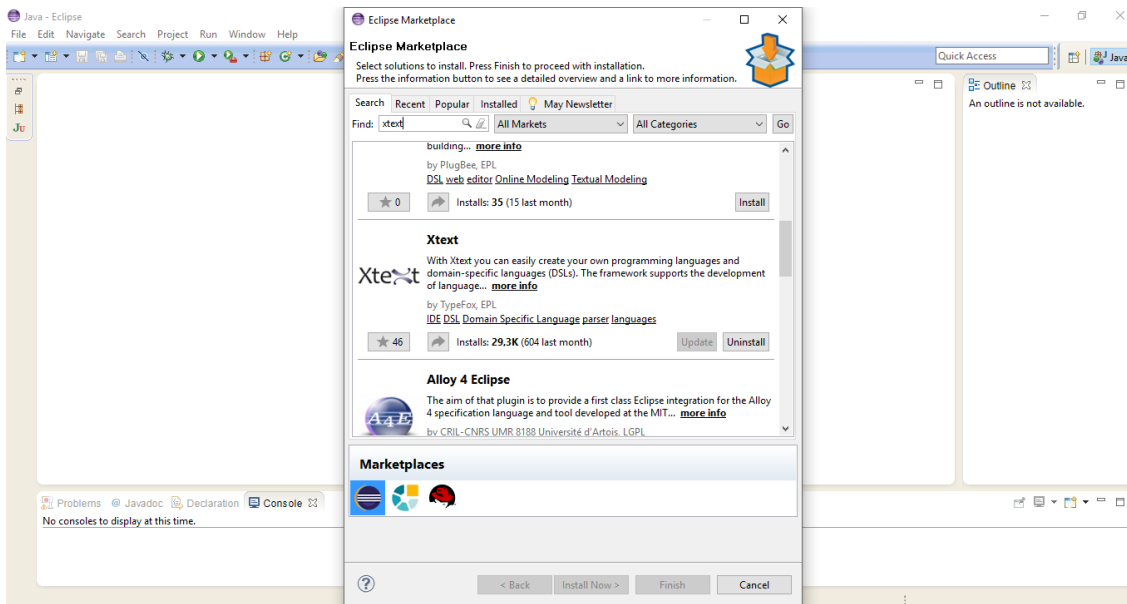


Ilustración 91 Eclipse Marketplace con Xtext

Después de instalarlo, se reiniciará el eclipse y ya estará listo para importar el proyecto generador. Cuando lo tengamos importado, podemos ejecutarlo pulsando con el segundo botón en la primera de las carpetas que se muestran y seleccionando *Run as -> Eclipse Application*.

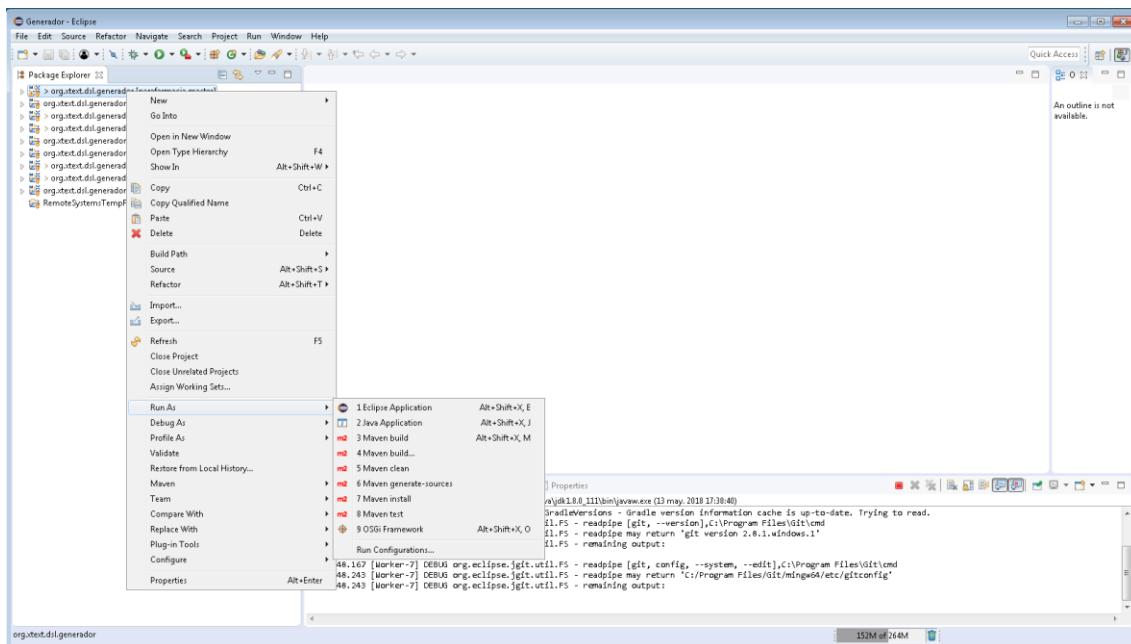


Ilustración 92 Eclipse Xtext

Después de ejecutarlo se abrirá una nueva ventana de eclipse con varios archivos .gen que permitirán introducir el código de creación.

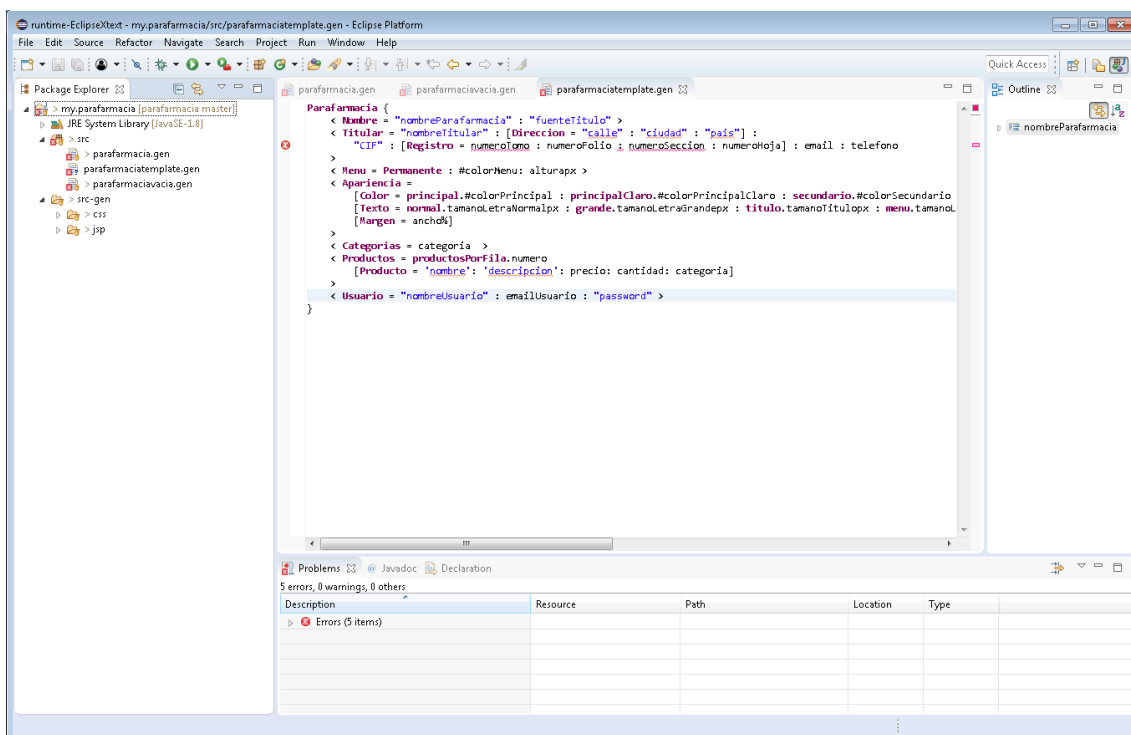


Ilustración 93 Eclipse generador Xtext

9.2 Manual de Usuario

A continuación, se van a realizar tres manuales de usuario diferentes para cada uno de los subsistemas del proyecto.

9.2.1 Parafarmacia Web

El manual de usuario de la parafarmacia web se va a subdividir en diferentes subtareas basadas en las historias de usuario.

Cuando entramos por primera vez en la parafarmacia nos aparecerá la siguiente pantalla:

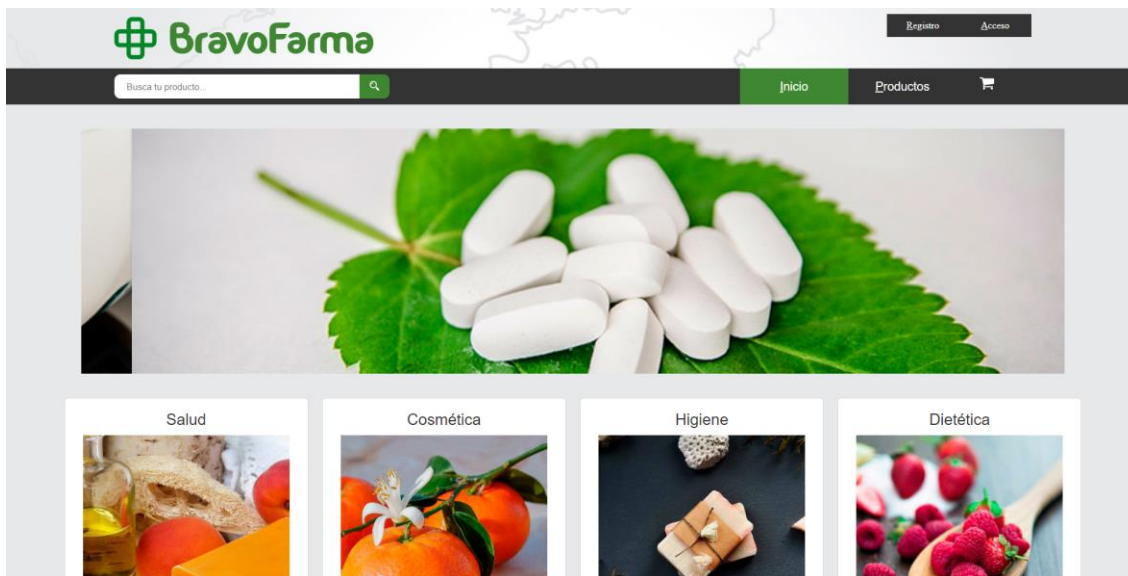


Ilustración 94 Pantalla principal parafarmacia web

9.2.1.1 Registro

Para registrarnos en la aplicación debemos pinchar sobre el enlace de “Registro” situado en la parte superior derecha de la pantalla:

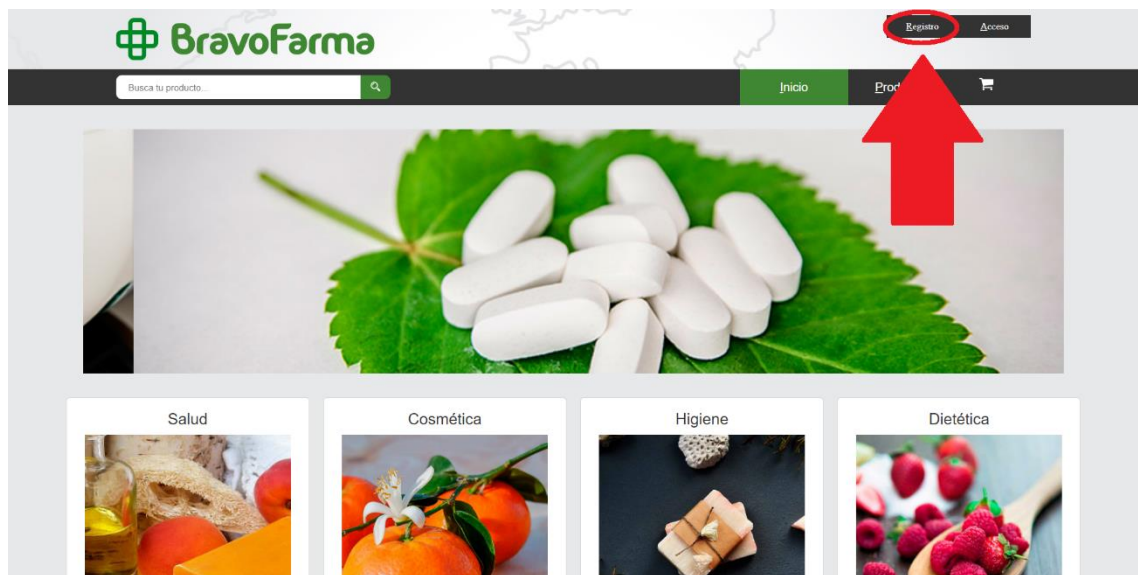


Ilustración 95 Enlace de registro parafarmacia web

A continuación, se abrirá una nueva pantalla en la que veremos un formulario con diferentes campos a rellenar para completar el registro:

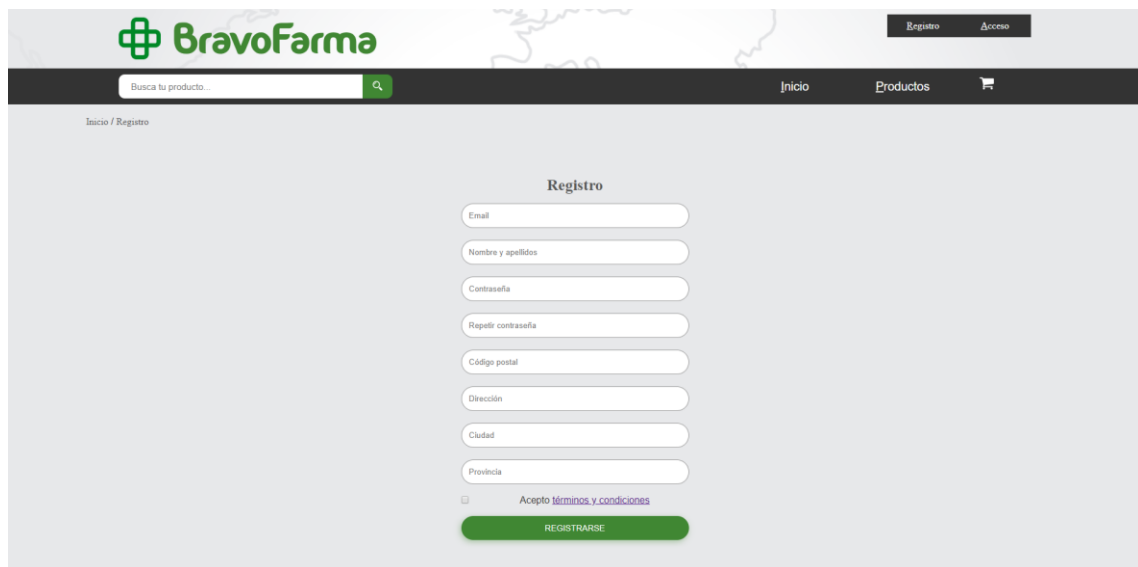


Ilustración 96 Pantalla de registro parafarmacia web

Introducimos nuestros datos, la contraseña por duplicado, sin olvidarnos de aceptar los términos y condiciones y por último pinchamos sobre el botón “Registrarse”.

Registro

pablo_bravo3@hotmail.com

pablo bravo

.....

.....

33010

calle la estación

oviedo

asturias

Acepto [términos y condiciones](#)

REGISTRARSE




Ilustración 97 Formulario de registro parafarmacia web

9.2.1.2 Iniciar sesión

Para iniciar sesión en la parafarmacia debemos seleccionar el enlace “Acceso” que se encuentra en la parte superior derecha:

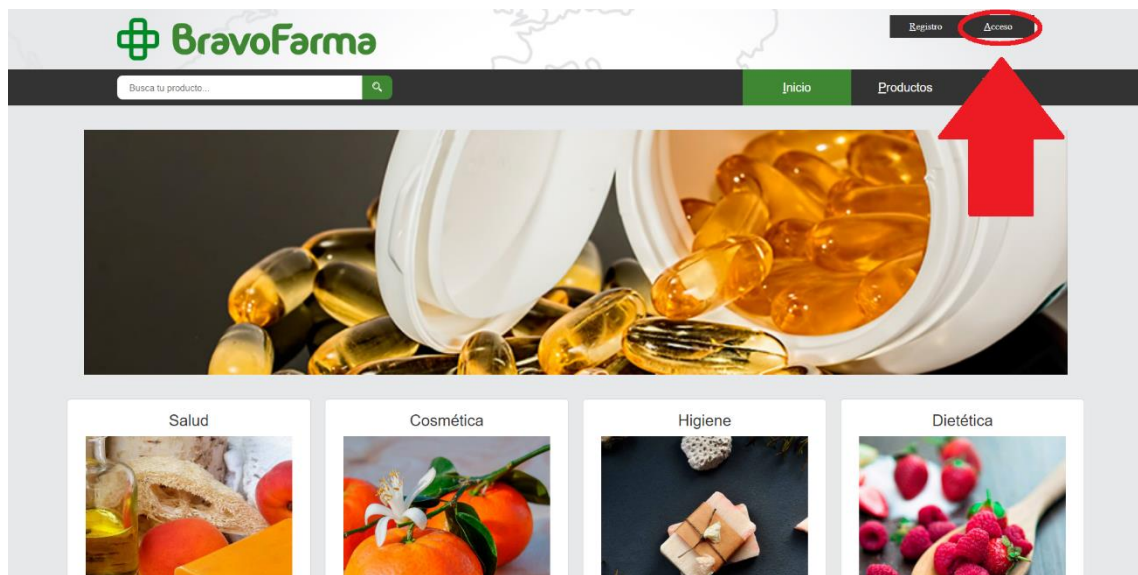


Ilustración 98 Enlace de acceso parafarmacia web

Al pinchar sobre el enlace se nos abrirá una nueva vista en la que aparece un formulario en el que deberemos introducir el email o nombre de usuario y la contraseña:

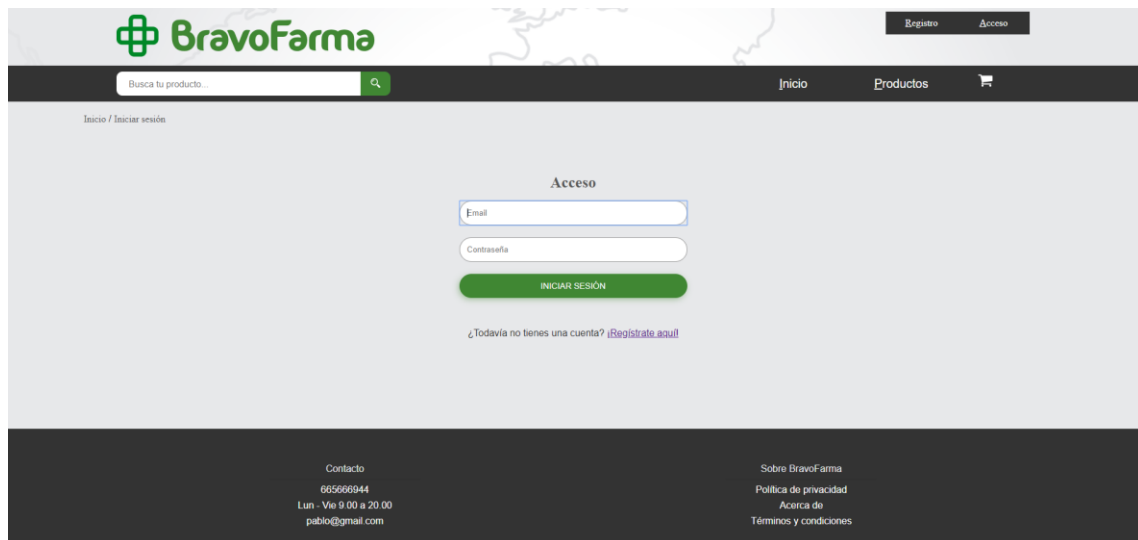


Ilustración 99 Acceso parafarmacia web

Una vez introducimos nuestros datos, pinchamos sobre el botón “Iniciar sesión”.

9.2.1.3 Cerrar sesión

Para cerrar sesión, en primer lugar, debemos haber iniciado sesión previamente, si es así, nos aparecerá en la parte superior derecha un enlace que pone “Cerrar sesión”, solo debemos pinchar en él y nuestra sesión finalizará.



Ilustración 100 Cerrar sesión parafarmacia web

9.2.1.4 Modificar cuenta

Para modificar la cuenta, en primer lugar, debemos iniciar sesión como ya hemos visto antes. Con la sesión iniciada nos aparecerá un enlace en la parte superior derecha que pone “Cuenta”, dicho enlace se desplegará al poner el ratón encima y aparecerá un submenú con la opción “Modificar”, pulsando sobre ella iremos a la nueva vista.



Ilustración 101 Modificar cuenta parafarmacia web

Como podemos ver en la siguiente imagen, el formulario que se muestra contiene todos los datos de nuestra cuenta, ahí podremos editar cualquiera de los valores y pulsar sobre el botón “Actualizar” para cambiarlos.

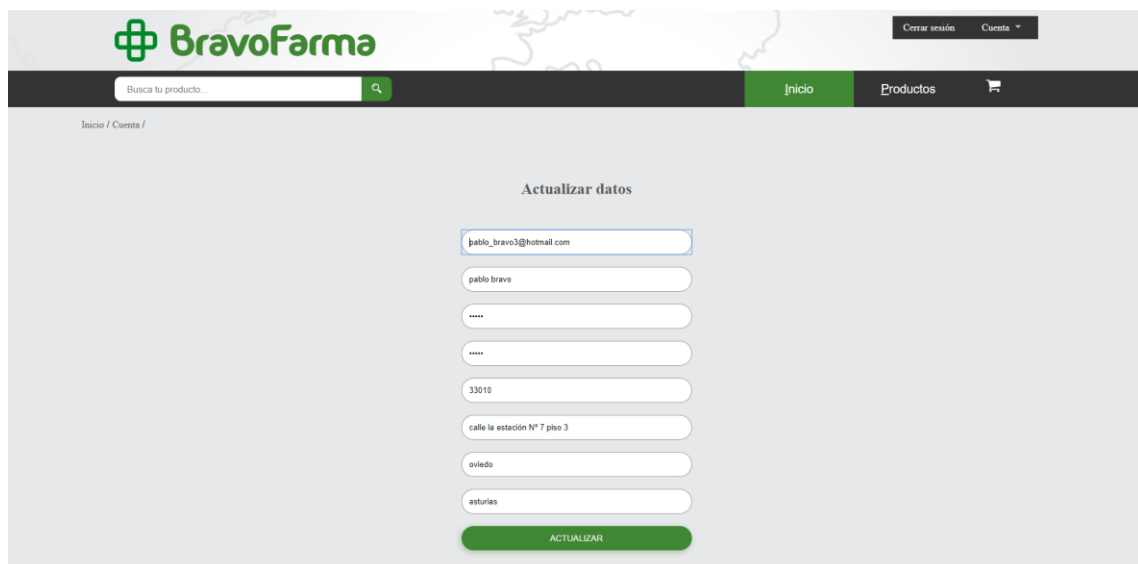


Ilustración 102 Formulario para modificar cuenta parafarmacia web

9.2.1.5 Comprar

Para realizar la compra de uno o varios productos, tenemos la opción ver todos los productos disponibles, filtrarlos por categoría o buscarlos por alguna palabra de su nombre.

Para ver todos los productos pincharemos en el menú sobre el enlace “Productos”:

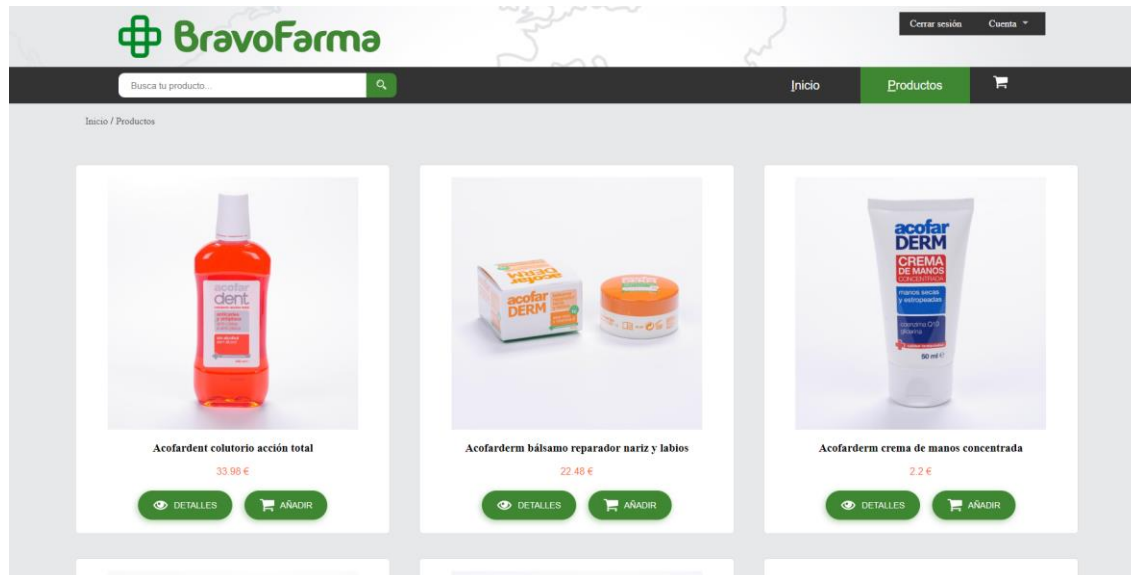


Ilustración 103 Productos parafarmacia web

Si quisiéramos buscar por el nombre, lo escribiríamos en el formulario que aparece en la barra de menú.

Para añadir productos al carrito pulsaremos sobre el botón “Añadir” que aparece en cada uno de los productos y automáticamente el carrito se actualizará mostrando el número de productos que hemos añadido.

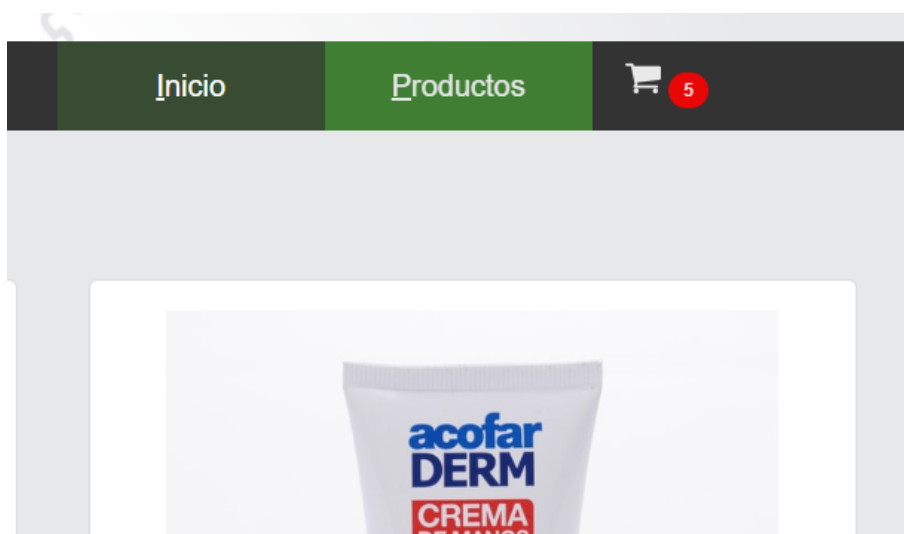


Ilustración 104 Número productos carrito parafarmacia web

Una vez que tengamos añadidos todos los productos que queramos comprar, pulsamos sobre el carrito que aparece en el menú:

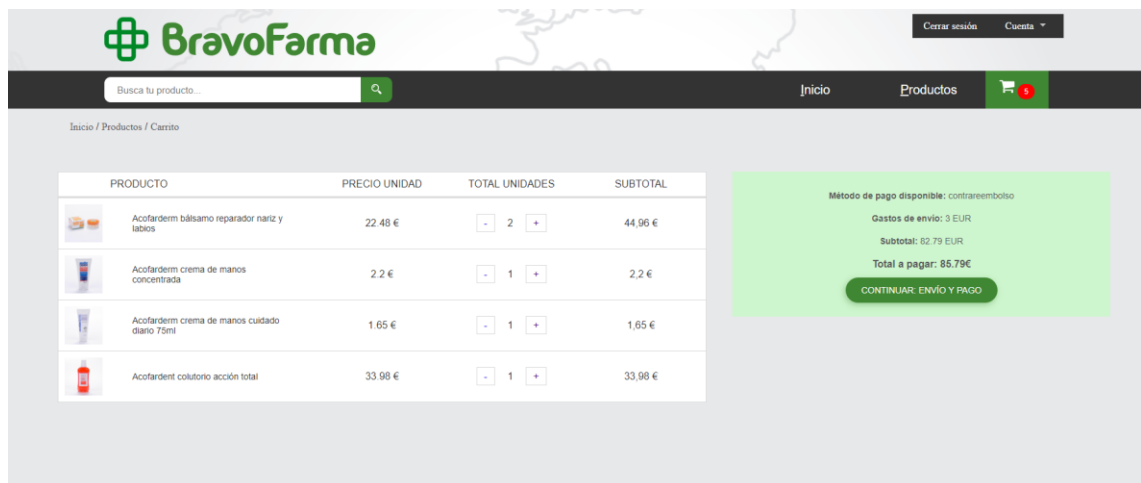


Ilustración 105 Carrito parafarmacia web

Podemos ver que nos aparece una lista con los productos que hemos añadido y la posibilidad de añadir o restar unidades de cada uno.

Para continuar, pincharemos sobre el botón “Continuar: Envío y pago”, nos aparecerá una vista con el resumen de la compra y un botón para finalizar.

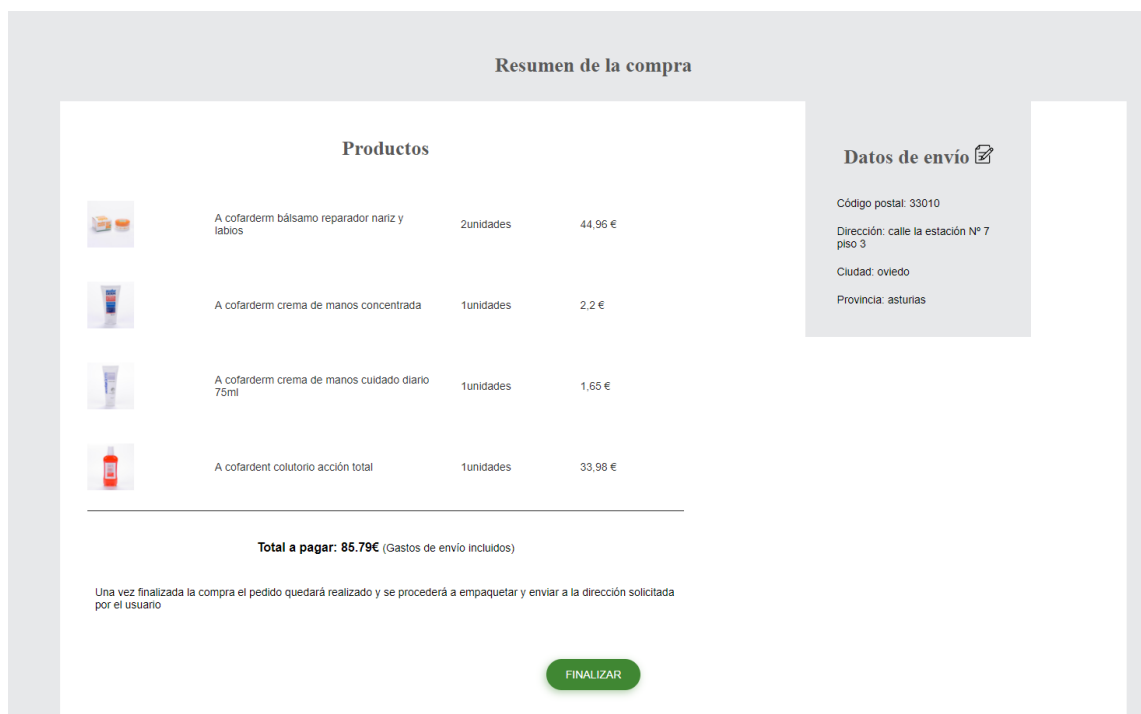


Ilustración 106 Resumen compra parafarmacia web

Al pulsar sobre “Finalizar”, la compra se habrá realizado y se mostrará un mensaje final:



Ilustración 107 Compra finalizada parafarmacia web

9.2.1.6 Ver pedidos

Para ver los pedidos que hemos realizado, en primer lugar, será necesario iniciar sesión en la aplicación. A continuación, pondremos el ratón encima del enlace “Cuenta” que aparece en la parte superior derecha, entonces se desplegarán dos opciones, la primera de ella, será “Mis pedidos”, pinchando sobre ella se nos abrirá una nueva vista con la lista de pedidos que hemos realizado:

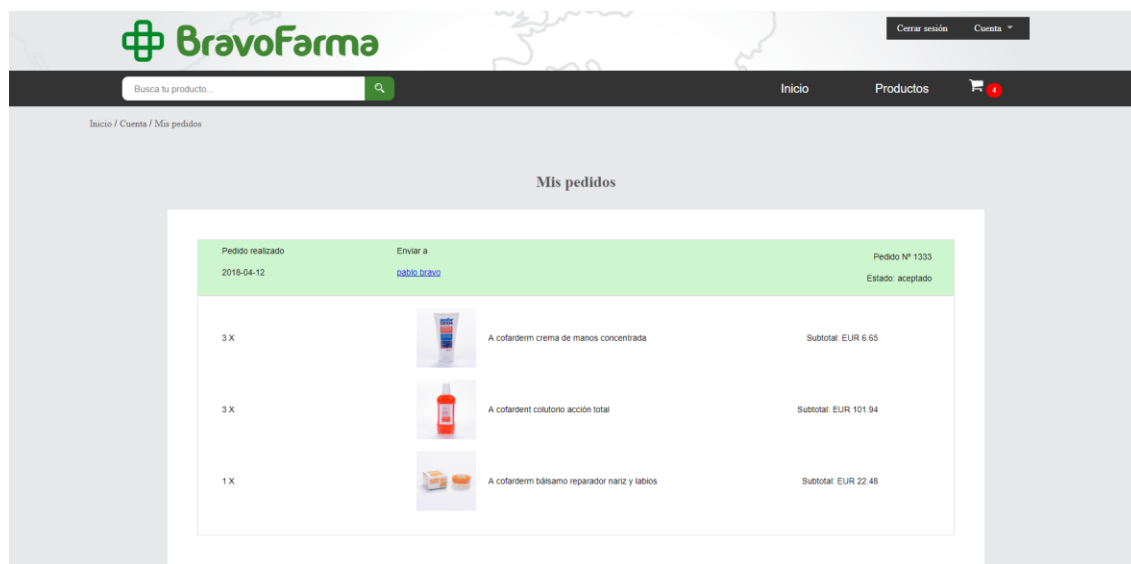


Ilustración 108 Mis pedidos parafarmacia web

9.2.2 Cliente Android

En este apartado se explicarán las diferentes tareas que se pueden realizar con el cliente Android. A continuación, se muestran la pantalla de inicio que dura cinco segundos y se quita sola y la pantalla principal:

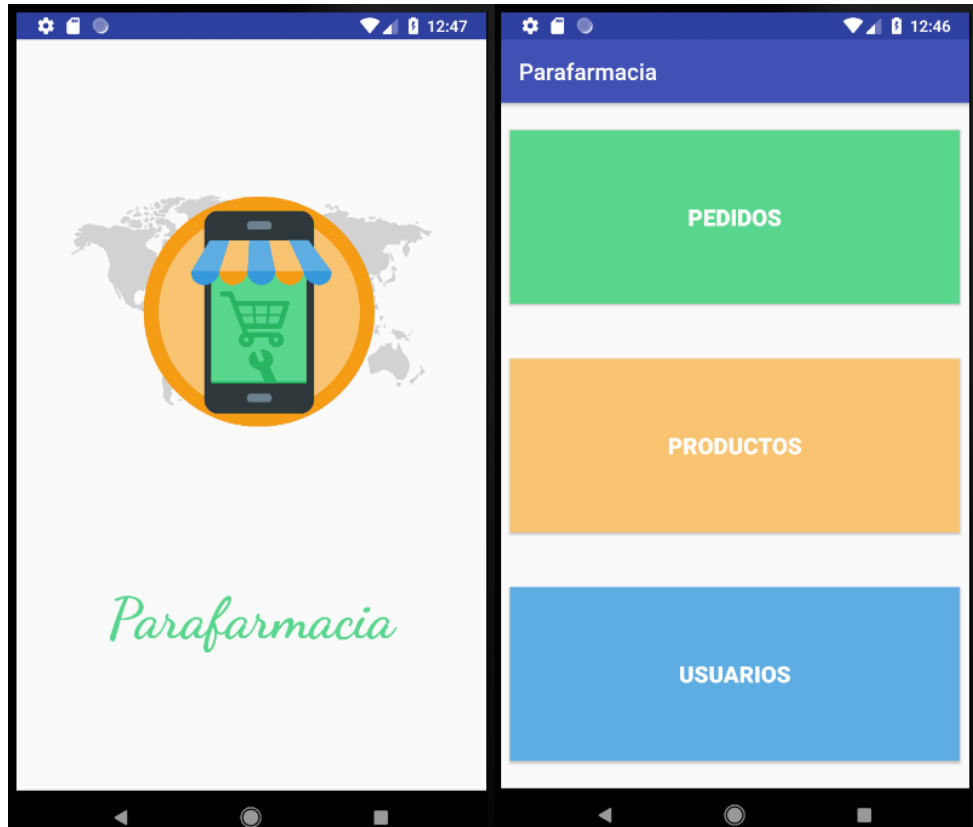


Ilustración 109 Pantalla de inicio y principal cliente Android

9.2.2.1 Ver pedidos

Para ver los pedidos que han realizado los clientes de nuestra parafarmacia web, solo debemos pinchar sobre el botón “Pedidos” que aparece en la pantalla principal.



Ilustración 110 Pantalla principal con opción pedidos cliente Android

Una vez pinchado, se abrirá una nueva vista con la lista de pedidos:

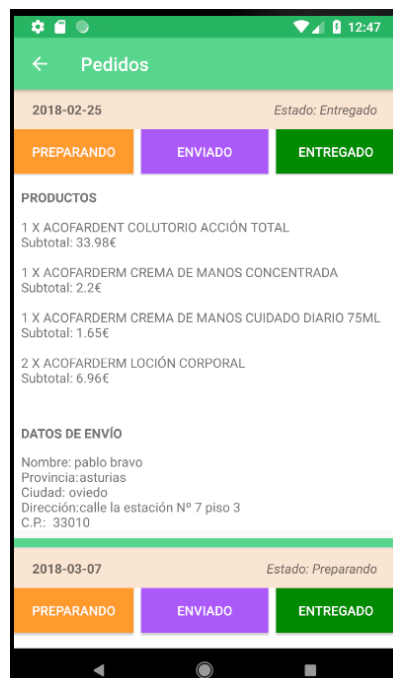


Ilustración 111 Pantalla pedidos cliente Android

9.2.2.2 Actualizar pedido

Para actualizar un pedido, en primer lugar, tenemos que ir a la vista de pedidos, en ella, podemos ver que para cada pedido tenemos tres botones diferentes, “Preparando”, “Enviado” y “Entregado”, si pulsamos sobre uno de ellos, el pedido modificará su estado en función del botón que elegimos.



Ilustración 112 Pantalla pedidos con estado cliente Android

9.2.2.3 Ver usuarios

Para ver los usuarios de la parafarmacia debemos estar en la pantalla principal, a partir de ahí simplemente seleccionaremos el botón de “Usuarios”:

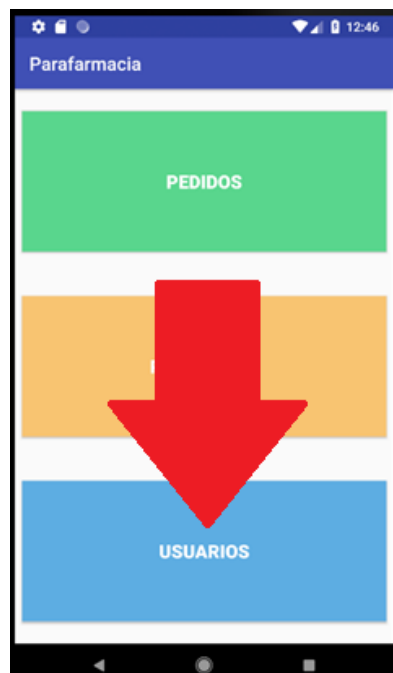


Ilustración 113 Pantalla principal con opción usuarios cliente Android

Una vez pulsemos el botón se abrirá una nueva vista con la lista de los usuarios:

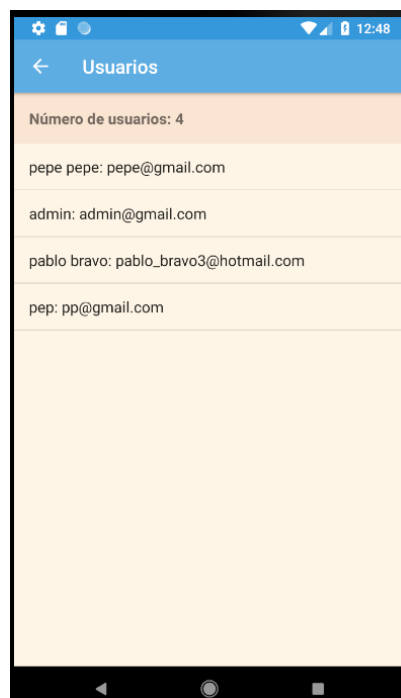


Ilustración 114 Pantalla usuarios cliente Android

9.2.2.4 Ver productos

Para ver los productos que tenemos en nuestra parafarmacia, debemos partir de la pantalla principal, en ella seleccionaremos el botón que aparece situado en el medio, "Productos":



Ilustración 115 Pantalla principal con opción productos cliente Android

A continuación, se mostrará una nueva vista con la lista de los productos:



Ilustración 116 Pantalla productos cliente Android

9.2.2.5 Añadir producto nuevo

Para añadir un nuevo producto a nuestra parafarmacia, debemos ir a la pantalla de productos, en ella veremos un botón “Nuevo producto” situado en la parte inferior de la pantalla:



Ilustración 117 Pantalla productos con nuevo producto cliente Android

Al pinchar sobre el botón, se abrirá una nueva vista que tendrá un formulario para rellenar con los datos del nuevo producto, nombre, descripción, precio, disponibilidad y categoría. Una vez tengamos todos los campos completados, pulsaremos sobre el botón “Añadir” que aparece en la parte inferior:

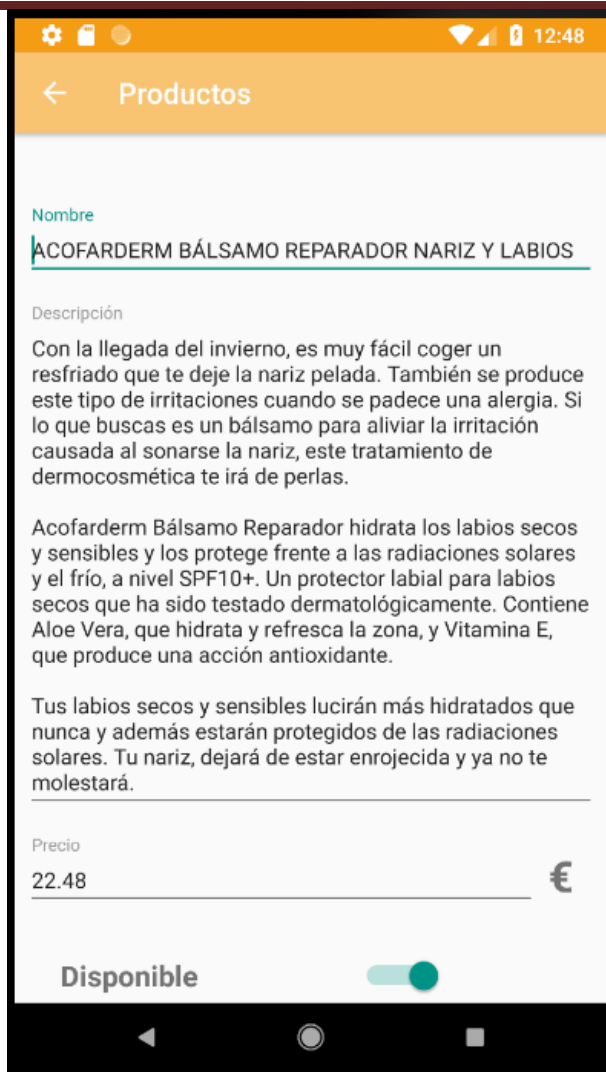


Ilustración 118 Pantalla nuevo producto cliente Android

9.2.2.6 Actualizar producto

Para actualizar un producto, debemos ir a la pantalla de productos y buscar en la lista el producto que queramos actualizar. Una vez lo tengamos visible, pulsaremos sobre el botón “Actualizar” que se encuentra al lado del producto:

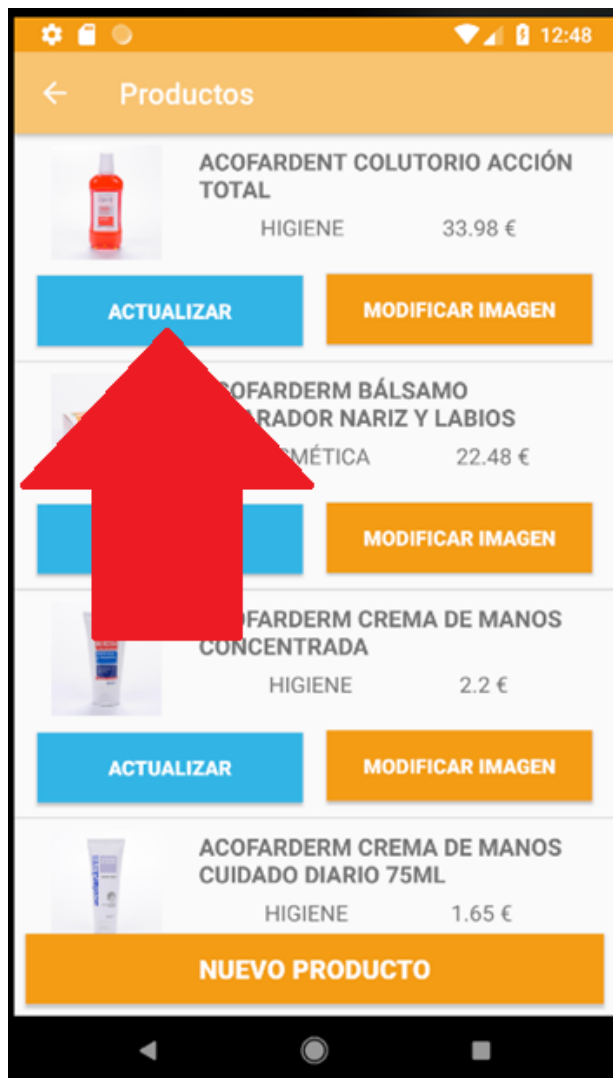


Ilustración 119 Pantalla productos con opción actualizar producto cliente Android

Una vez pulsemos sobre él, se abrirá una nueva vista similar a la de añadir productos con los datos del producto seleccionado ya introducidos en el formulario. Podemos seleccionar uno de los campos y editar el texto. Para finalizar, tenemos que darle al botón que aparece en la parte inferior “Actualizar”.

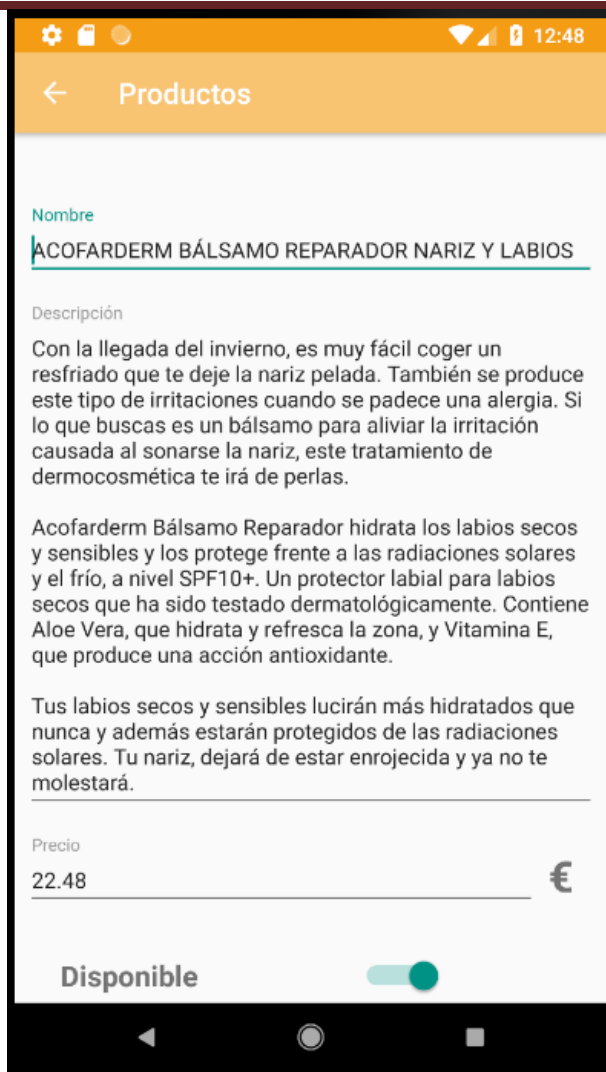


Ilustración 120 Pantalla actualizar producto cliente Android

9.2.2.7 Generador

Para poder generar una nueva parafarmacia, se debe escribir un nuevo archivo con el lenguaje que se explicará a continuación. Este archivo no tendrá extensión “.txt” si no que será “.gen”. Para que la parafarmacia se genere, el archivo debe estar escrito sin ningún fallo y solamente con guardar su contenido, crearemos todo lo necesario. Si se produce algún fallo el programa nos avisará con una cruz roja en la línea del fallo indicando el error si nos situamos encima con el ratón.

Ahora, se procederá a describir cada una de las partes del lenguaje. Como estamos creando una parafarmacia, es lógico pensar que el código comience con una palabra clave “Parafarmacia” que recogerá el resto de partes que la forman. Dentro de la parafarmacia tendremos varias opciones a personalizar, el nombre de la tienda, el titular, el usuario principal, las categorías junto a los productos y la apariencia.

El esquema básico del lenguaje utilizado es el siguiente: La parafarmacia encierra a todas sus partes mediante llaves { ... }, cada una de las partes (apariencia, productos, categorías...) se encierra mediante los símbolos de mayor y menor < Apariencia ... >, las partes contienen los datos que podemos personalizar, estos datos irán separados por dos puntos < Apariencia = dato1 : dato2 >, por último, tenemos los datos complejos, que son aquellos formados por varios subdatos, por ejemplo, la apariencia de la parafarmacia se divide en colores, textos y márgenes, estos se consideran datos complejos ya que tenemos diferentes zonas con diferentes colores, color de fondo, color de productos, color de textos, color de menú... Los elementos complejos funcionan igual que los simples, pero se encierran en corchetes [Color = color1 : color2].

Por suerte, escribir el código para generar la parafarmacia es muy fácil, solo es necesario pulsar la combinación de teclas Ctrl + espacio y se desplegará un cuadro de diálogo con una ayuda para escribir el código. Esta ayuda contiene el esqueleto completo y solo será necesario escribir dentro de los recuadros que se mostrarán. Para facilitar más aun las cosas, para pasar de un dato a otro y empezar a escribir en el momento, solo tendremos que pulsar sobre la tecla del tabulador ↹.

9.3 Manual del Programador

9.3.1 Parafarmacia Web

La parafarmacia web se ha desarrollado utilizando el framework Spring, la arquitectura seguida es modelo vista controlador, si necesitamos añadir nuevas vistas a la parafarmacia será necesario crear un controlador nuevo que recoja la petición y devuelva la nueva vista solicitada.

Para añadir nuevas entidades al modelo con sus correspondientes accesos a la base de datos, es necesario saber tres cosas básicas:

1. Se ha utilizado un patrón DAO que permite conectar nuevas bases de datos.
2. Es necesario declarar la clase del modelo junto a las clases de acceso a la base de datos en el archivo de configuración que contiene los beans.
3. Existe un archivo de configuración que contiene los datos para conectarse con la base de datos y otro con las consultas.

Para las vistas, debemos tener en cuenta que tenemos dos archivos generales, el menú y el footer, que se incluyen en cada una de las vistas.

9.3.2 Cliente Android

El cliente Android está desarrollado utilizando JAVA, algunos detalles que debemos saber para poder realizar cambios en él son:

- La pantalla de inicio o splash, contiene los métodos asíncronos que permiten la conexión con la base de datos de la parafarmacia a través de los servicios web SOAP para obtener una copia de los datos y almacenarlos en la base de datos SQLite interna.
- Para crear un nuevo Layout en la aplicación es necesario inicializarlo a partir de su Activity.
- Las listas personalizadas que aparecen con los productos y los pedidos se han creado mediante el uso de adaptadores que permiten distribuir elementos a nuestro gusto en un nuevo Layout y mostrar cada uno de ellos como una fila de la lista.
- Si queremos añadir un nuevo método que se conecte a la base de datos de la parafarmacia para añadir o modificar contenidos, será necesario conectarse con el servicio web con la librería KSOAP2 importada en el proyecto.
- Para crear una nueva tabla en la base de datos hay que modificar la clase MYDBHelper y añadir la consulta de creación.
- La forma de actualizar las imágenes de un producto es mediante un WebView que contiene código PHP que permite insertar la imagen directamente en la carpeta de productos de la parafarmacia.
- La carga de las imágenes de los productos en la lista se realiza mediante la librería Picasso que permite almacenar las imágenes en caché.
- Para subir imágenes a la parafarmacia web se carga un formulario mediante un webview que contiene el código PHP encargado de realizar la tarea.

9.3.3 Generador

Aspectos a tener en cuenta a la hora de hacer ampliaciones en el generador:

- Si vamos a añadir nuevos elementos al lenguaje será necesario modificar la plantilla con el template del lenguaje para el usuario final.
- Si queremos generar nuevos archivos debemos crear un nuevo generador e inicializarlo desde la clase `GeneradorGenerator`.
- Cada vez que modifiquemos alguna parte del lenguaje creado deberemos generar el nuevo dominio.
- Para ayudar al usuario con nuevas partes añadidas en el lenguaje se pueden crear nuevas excepciones con mensajes de error personalizados.
- La clase que contiene el esquema del lenguaje se llama `DomainModel.xtext`.

Capítulo 10. TFM VS TFG

10.1 Parafarmacia Web

Wordpress vs Spring MVC

Wordpress es un gestor de contenidos que permite crear aplicaciones web de manera rápida, fácil y sin tener conocimientos de programación. Todo esto se consigue instalando diferentes plugins que añaden funcionalidad a nuestra aplicación, dichos plugins, pueden ser creados por cualquier usuario. El principal problema que tenemos es la personalización, Wordpress ofrece temas para cambiar la apariencia de nuestra aplicación, los que son gratuitos, son muy básicos y ofrecen un aspecto bastante pobre, además es complicado personalizarlos, quedando un código muy feo. Otro aspecto a tener en cuenta, es la seguridad, cuando instalamos el Wordpress en un servidor, ofrece mucha información que puede ser utilizada por posibles atacantes de la web. Si nos fijamos en el funcionamiento interno, vemos que las páginas cargan todo mediante su base de datos, lo que se traduce en un montón de consultas, lo que provoca un mayor consumo y menor rendimiento. Por último, el mayor problema que tenemos es que no controlamos su desarrollo, así que la conexión entre subsistemas, para este proyecto, es menos segura y menos estándar, ya que en el proyecto fin de grado se tuvieron que desarrollar complementos y pseudo servicios web que pese a funcionar correctamente, no eran una solución con el nivel adecuado de profesionalidad.

Spring MVC es un framework que nos permite desarrollar mediante JAVA aplicaciones web, como Wordpress, tiene sus ventajas y desventajas, lo bueno que tenemos ahora es que los problemas que había con Wordpress, ahora están controlados y se pueden solventar con líneas de código o alguna configuración. La principal razón por la que se ha escogido este framework es que es mucho más compatible y una solución más adecuada para todas las necesidades del proyecto. Empezando por el generador, Wordpress no tiene una estructura clara de código que poder externalizar y personalizar, si no que todo lo que permite personalizar se encuentra almacenado en su base de datos, estilos de las páginas, textos, títulos, bloques, menú... así que, el generador no se podía considerar un generador como tal ya que lo único que hacía era modificar datos en el script de la base de datos. Ahora, con Spring, el código está claramente definido, la estructura se puede externalizar y personalizar al 100%, por lo que el generador es mucho más completo, permite crear diferentes vistas, generar la hoja de estilos, hacer uso de servicios web para generar productos directamente en la base de datos e incluso podría generar controladores que añadieran mayor funcionalidad a la web así como nuevos modelos de datos.

Pantalla principal

A continuación, se muestran las imágenes de las dos pantallas principales de la parafarmacia, como se puede ver, la antigua usa colores oscuros y simplemente muestra el nombre y un logo, mientras que la nueva página, utiliza colores claros, que permiten al usuario descansar la vista, muestra de manera global el escaparate de la tienda con las diferentes categorías de productos que podemos encontrar en ella y utiliza imágenes vistosas para acompañar y reforzar los textos así como transmitir la idea principal.



Ilustración 121 Pantalla principal Parafarmacia TFG

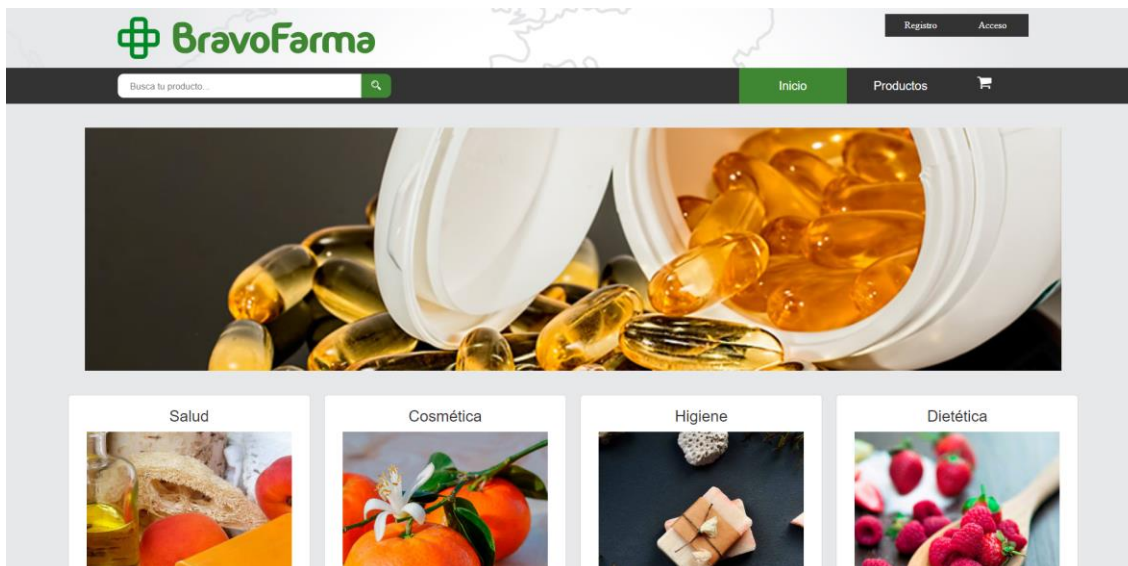


Ilustración 122 Pantalla principal Parafarmacia TFM

Identificador del sitio

Un aspecto fundamental de las páginas web es la relación que tienen con su marca, nombre y logo. Un usuario que acceda a una tienda online, debe poder identificar en todo momento a la empresa para la que va a realizar la compra, por lo que debemos tener claramente visible en nuestra web, en todas las vistas que la compongan, el logo y nombre por el que ser reconocidos. En la parafarmacia desarrollada para el TFG, vemos que únicamente tenemos visible un logo, que no es ni si quiera el mismo logo mostrado en la página principal, mientras que, en la nueva versión, tenemos siempre visible en la parte superior izquierda, con letras personalizadas y color destacable el nombre y logo de la parafarmacia. Además, esto permanece en el mismo lugar durante toda la navegación por la web.



Ilustración 123 Logos del TFG y TFM

Carrito con unidades

Otra mejora que hace la web más accesible es mostrar en todo momento el número de unidades que el usuario ha añadido al carrito. Como podemos ver en la primera imagen del TFG, pese a tener dos productos añadidos al carrito, el usuario no tiene ninguna constancia de ello mientras que en la nueva versión, el usuario sabe claramente que ha añadido cuatro productos.



Ilustración 124 Carrito del TFG y TFM

Menú desplegable

Con el objetivo de no saturar a los usuarios de información, es necesario mostrar en nuestra página la información necesaria y dejar que el usuario vaya descubriendo el resto, por ello, otra de las mejoras implementadas ha sido esconder las categorías de productos del menú, para que el usuario simplemente tenga las opciones básicas y añadir como desplegable las categorías de los productos. Además, para facilitar al usuario la navegación, se han separado los enlaces que tienen relación entre sí para dar mayor consistencia y que se pueda diferenciar y encontrar mucho más fácil. Por ello, se ha creado un submenú en la parte superior derecha para las opciones relacionadas con la cuenta de usuario, iniciar sesión, registrarse, cerrar sesión o modificar datos.

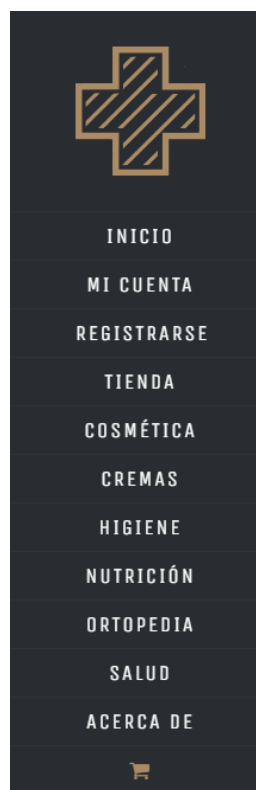


Ilustración 125 Menús del TFG y TFM

Migas

Otro aspecto importante que permite guiar al usuario por nuestra web, es el uso de las conocidas migas de pan, que muestran el recorrido de las páginas enlazadas que ha ido recorriendo el usuario para llegar a la actual. Además, permite al usuario seleccionar cada una de ellas y volver atrás fácilmente.



Ilustración 126 Uso de migas en el TFM

Colores

Cuando entramos en una página con el fondo oscuro y la letra clara, nos cuesta más leer los textos y tardamos más en hacerlo, por esto, en la nueva versión, se ha tratado de utilizar colores claros en todo momento y letras que resalten diferentes aspectos. Como se puede ver en la primera imagen, el color de fondo del menú es similar al resto de la página, por lo que no es fácilmente reconocible, en la nueva versión, se ha utilizado un color totalmente opuesto para resaltar el menú y que el usuario pueda apreciarlo de un solo vistazo. Además, para mantener la consistencia, se ha utilizado el mismo color oscuro del menú para el pie de página. En este caso el uso de colores oscuros de fondo no es un problema ya que el menú y el pie no contienen grandes párrafos que perjudiquen la lectura del usuario.

Por último, se puede ver el uso de variantes de colores claros para tratar de resaltar y guiar al usuario en los pasos para realizar un pedido.

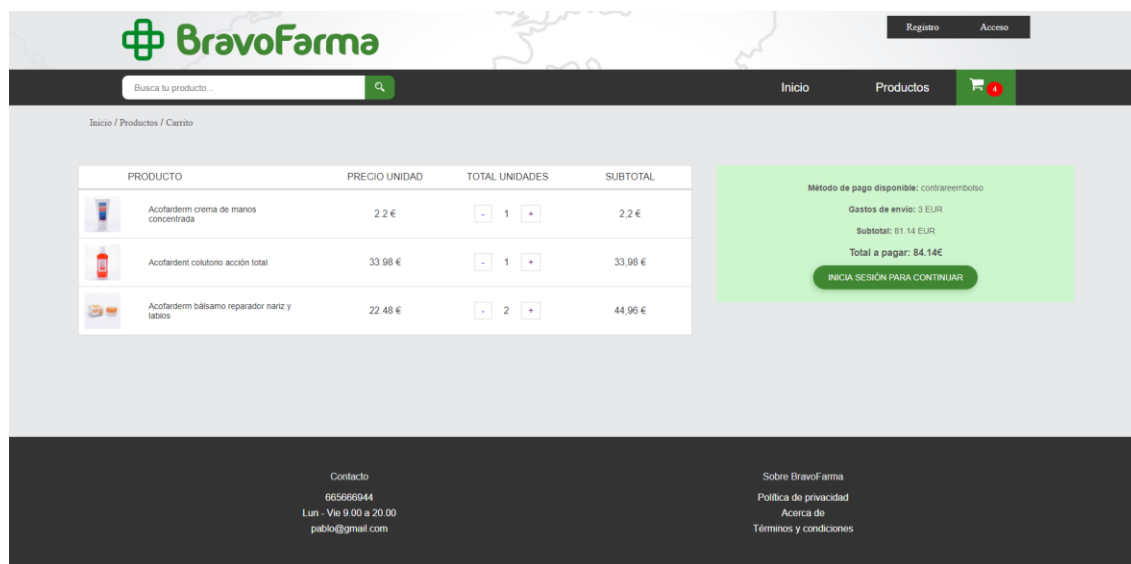


Ilustración 127 Contraste de colores TFG y TFM

Títulos

Los títulos de las páginas se utilizan para que el usuario sepa en qué página se encuentra, se pueden ver que en ambas páginas se utilizan, pero en la página del TFG se puede ver que tienen demasiada importancia, ocupan más del 50 por ciento de la página, reduciendo el espacio que dejamos al contenido, que casi queda desaparecido si no hacemos scroll. En la nueva página, podemos ver que siguen apareciendo, pero sin ocupar tanto espacio, dejando más visible el contenido y sin un tamaño de letra que reste importancia al identificador de la tienda.

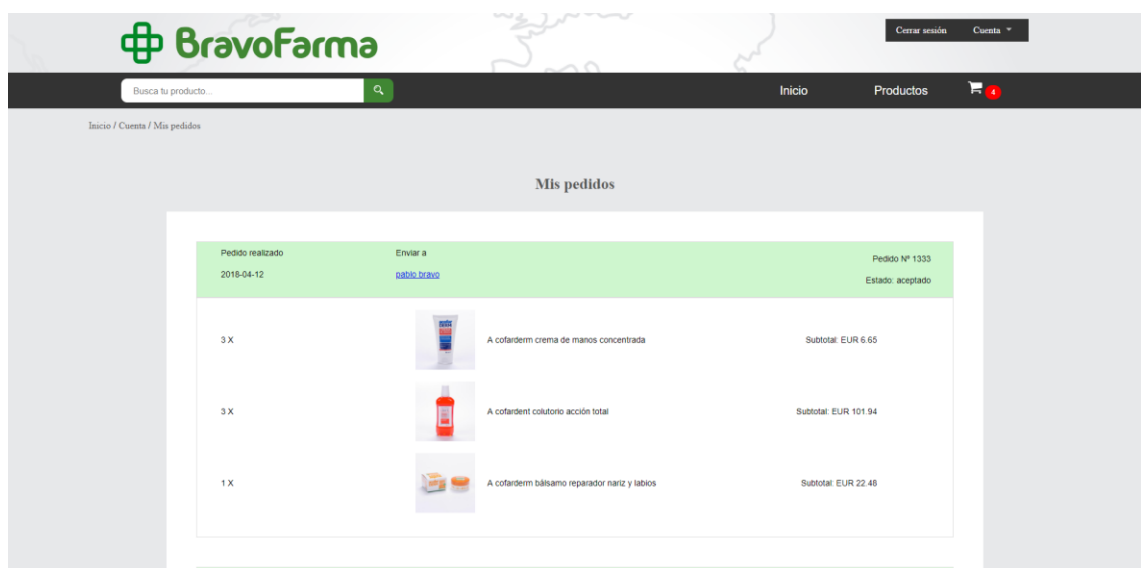


Ilustración 128 Títulos TFG y TFM

Formularios y uso de espacios

Es importante organizar la información por toda la página, como se puede ver en la primera imagen, tenemos toda la información en la primera mitad, dejando inutilizada la otra mitad, además, el botón de login tienen un tamaño muy pequeño, difícil de pinchar sobre él y poco reconocible. En la nueva versión, se ha tratado de centralizar el contenido, juntar los campos de texto del formulario con el fin de reducir el tiempo que el usuario tarda en pasar de uno a otro y aumentar el botón de login, haciendo que sea más fácil pinchar sobre él y situándolo justo debajo del último campo del formulario.

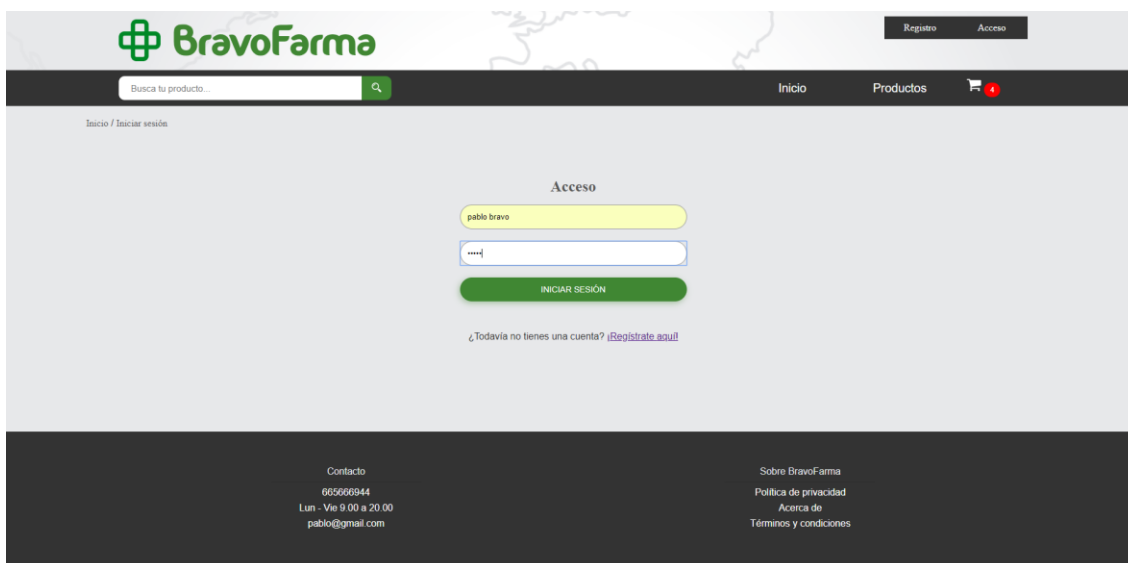
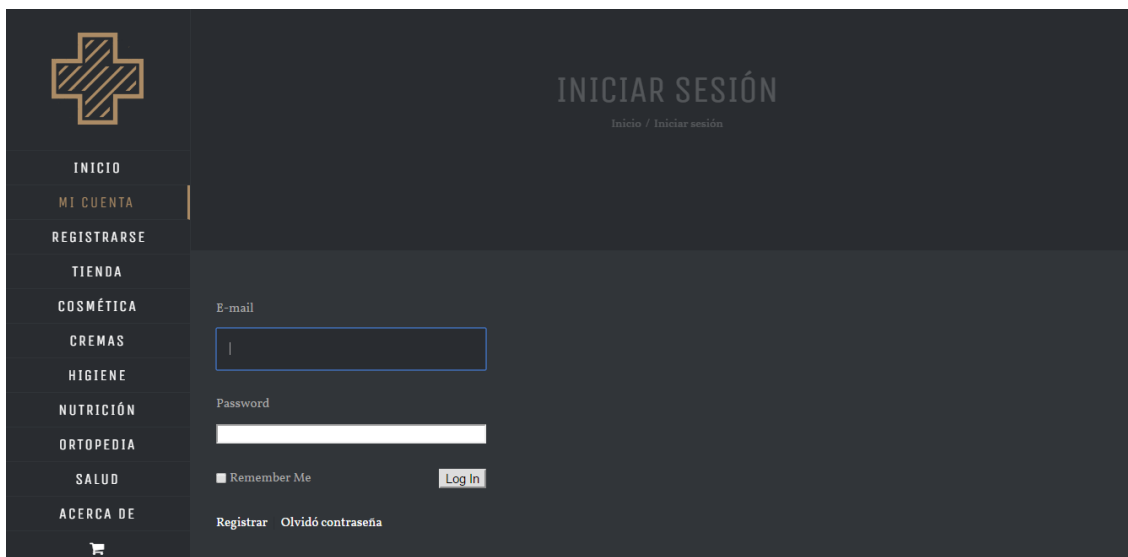


Ilustración 129 Formularios TFG y TFM

Atajos

Para facilitar a los usuarios la navegación por la web, se han incluido una serie de atajos que la antigua página no tenía. Como podemos ver, para que se reconozcan estos atajos, se ha subrayado la letra con la cual nos moveremos ha dicho enlace. Por ejemplo, si queremos ir a la página de registro, pincharemos alt+A para acceder rápidamente, para inicio, alt+I.

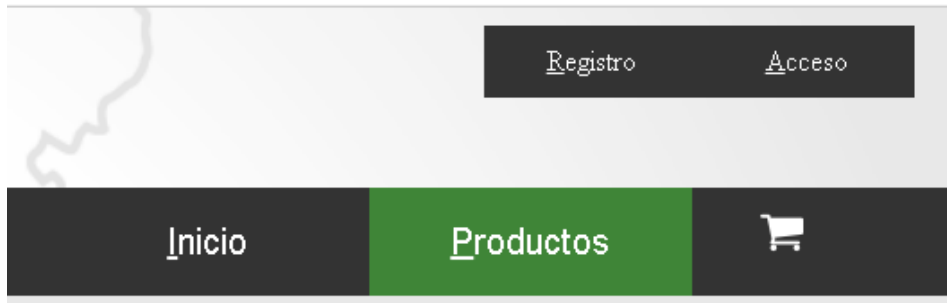


Ilustración 130 Atajos TFM

URL

Otra forma de guiar al usuario por la página es con URLs fáciles y significativas, por esto, se ha tratado de formar las URL lo más simples posibles, para ello se ha intentado en la medida de lo posible evitar el paso de variables por método GET para no tener URLs largas con muchas variables.

/productos

/carrito

/productos/detalles

/registro

Buscador

Toda tienda online necesita tener visible en todas sus páginas un buscador de productos que le permita en cualquier momento encontrar lo que necesite. Como vemos en la antigua web, únicamente teníamos la opción de ordenar los productos por alguna característica, pero no se podía buscar. En la nueva web se ha incluido en el menú un buscador que estará siempre disponible y que permite introducir cualquier palabra del nombre o descripción de un producto. Además, a modo de filtro, se puede buscar los productos por diferentes categorías.

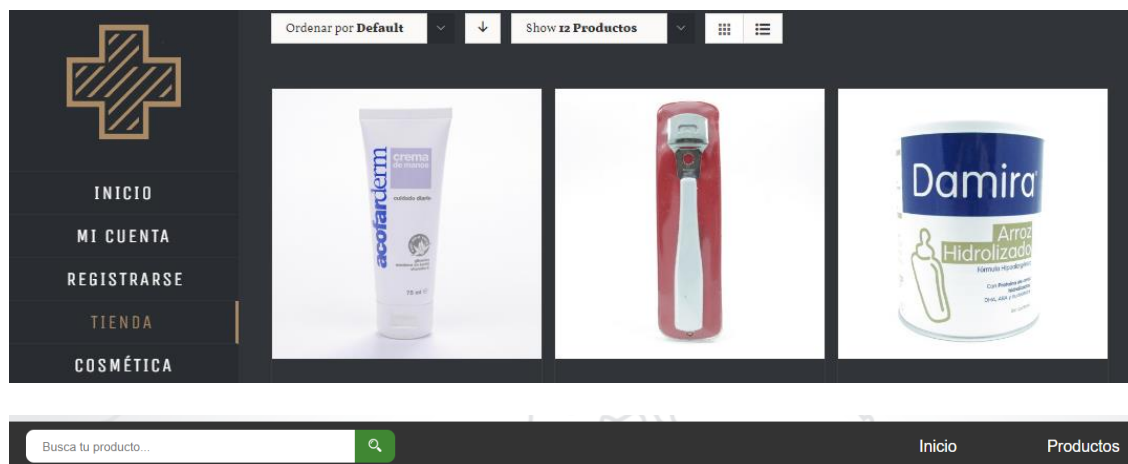


Ilustración 131 Buscador TFG y TFM

Mostrar actualizaciones

Otro añadido necesario para las pruebas de accesibilidad es añadir la fecha de la última actualización de la web. Esto en la antigua web no estaba, pero en la nueva se ha añadido, además, el generador, cada vez que crea una nueva parafarmacia, genera la fecha de ese día y la coloca en el pie de la parafarmacia.

Cookies

En la antigua versión de la parafarmacia no se habían tenido en cuenta ni se habían utilizado las cookies. En la nueva versión, se ha añadido el aviso obligatorio de cookies que recoge la LSSI, que se encuentra en la parte inferior de la página de forma fija, como se ve en la siguiente imagen:

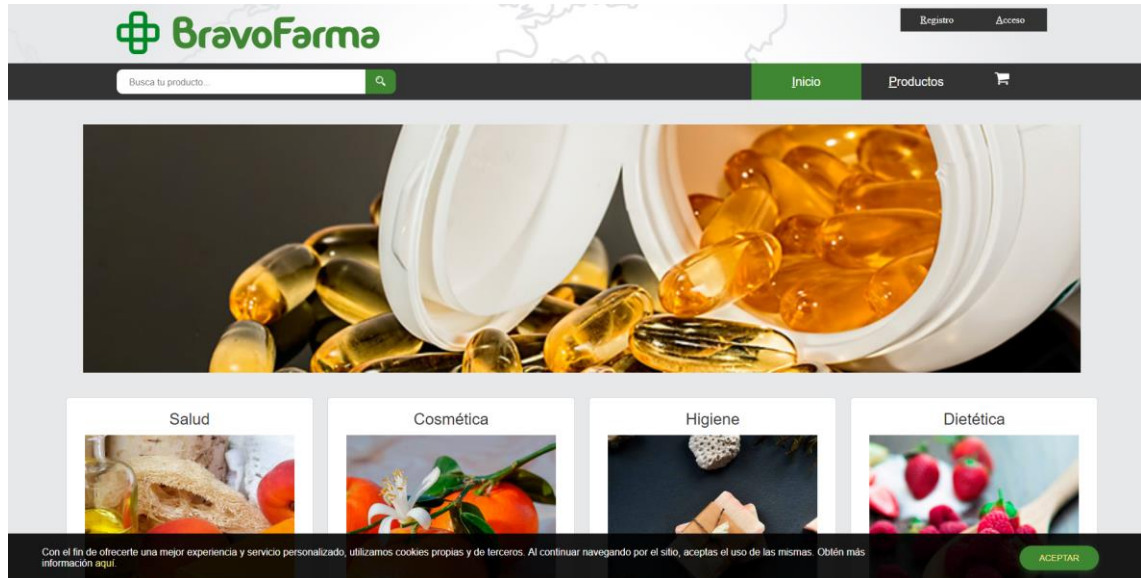


Ilustración 132 Página principal con cartel de cookies parafarmacia web

Además, dentro del aviso, tenemos un enlace que permite visualizar la política de cookies de la página:



Ilustración 133 Política de privacidad y cookies parafarmacia web

Por último, para facilitar el inicio de sesión de los usuarios a nuestra tienda, cuando un usuario accede por primera vez, se le guardan los datos de acceso en una cookie, de esta forma, cuando el usuario vuelva a iniciar sesión, los campos del formulario de acceso le aparecerán completos con sus datos, lo que le permite empezar a comprar más rápido.

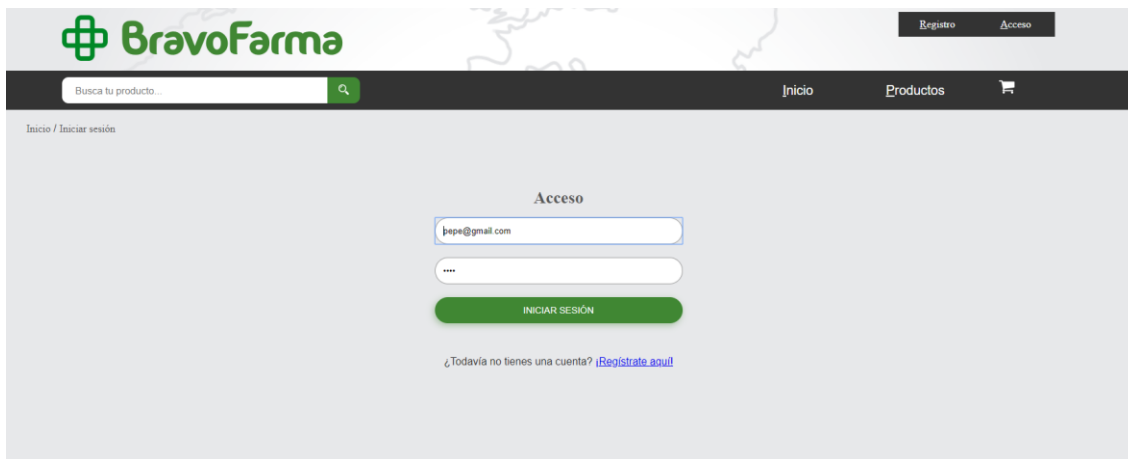


Ilustración 134 Inicio de sesión parafarmacia web con cookies

10.2 Servicios Web

Comenzando por el antiguo proyecto de TFG, los servicios web se tuvieron que desarrollar para conectar el cliente Android con la base de datos de Wordpress. Estos servicios, consistían en un archivo PHP almacenado junto al resto de archivos de Wordpress, que realizaban diferentes tareas, obtener productos, obtener pedidos... Cuando el cliente Android necesitaba obtener alguno de estos datos, realizaba una conexión HTTP a la URL del archivo PHP y este devolvía el código con los datos necesarios, por esta razón los llamo pseudo servicios web.

Para la nueva versión, se ha desarrollado un proyecto con JAVA que implementa los métodos encargados de acceder a la base de datos de la parafarmacia web y ofrece estos métodos como servicio web SOAP. Los subsistemas que consumen este servicio son el generador y el cliente Android. El generador, que en la versión antigua no tenía esta funcionalidad, ahora se conecta con el servicio para limpiar la base de datos y añadir los nuevos productos de la parafarmacia generada. El cliente Android, mediante la librería ksoap2, realiza una conexión con la base de datos para hacer una copia de ella, así como añadir, actualizar o borrar datos cuando sea solicitado. Consumir el servicio web desde Android no es una tarea fácil, ya que no tiene la opción de crear las clases de modelo para obtenerlos directamente al ejecutar los métodos, si no que el servicio web en este caso devuelve una única variable con un JSON que contiene la información devuelta por el método del servicio y es necesario transformarla al objeto correspondiente.

Como se puede ver, hay una gran diferencia entre el desarrollo antiguo en el que se intentaba hacer una imitación de un servicio y este nuevo desarrollo que tiene un servicio web SOAP implementado siguiendo los estándares.

10.3 Cliente Android

Antes de hablar de las diferencias entre la interfaz de usuario nueva y la antigua, se va a comentar la diferencia entre la eficiencia de la aplicación.

Al utilizar servicio web de verdad, el cliente Android permite, al abrir la aplicación, conectarse con la base de datos de la parafarmacia y actualizar u obtener los datos que hay en ella, esto servirá a modo de caché, para evitar múltiples conexiones cada vez que accedemos a una nueva vista, con la consecuente pérdida de tiempo, cosa que ocurría en la antigua versión, cada vez que abríamos una vista que necesitaba datos de la parafarmacia, realizaba conexiones para obtenerlos y a continuación los mostraba. Además, para aumentar el rendimiento, se ha incluido para la nueva versión el cacheado de imágenes mediante la librería "Picaso", que almacena en la memoria las imágenes y las carga cuando la vista las va solicitando, funcionando incluso sin tener una conexión a internet.

Ahora, pasaremos a comentar las diferencias en la interfaz.

Pantalla principal

Se puede ver como principal diferencia que la antigua pantalla contiene mucha información en la pantalla principal que el usuario, al ser el administrador de la parafarmacia, ya conoce y no necesita tener ahí. Además, las principales funciones se encuentran ocultas en el menú. La nueva pantalla ha eliminado toda la información innecesaria, es mucho más limpia y ha colocado las tres opciones principales fácilmente accesibles para el administrador. También se ha incluido, con el fin de guiar ha dicho administrador, diferentes colores para cada opción que se mantendrán consistentes al acceder a cada una de las opciones.

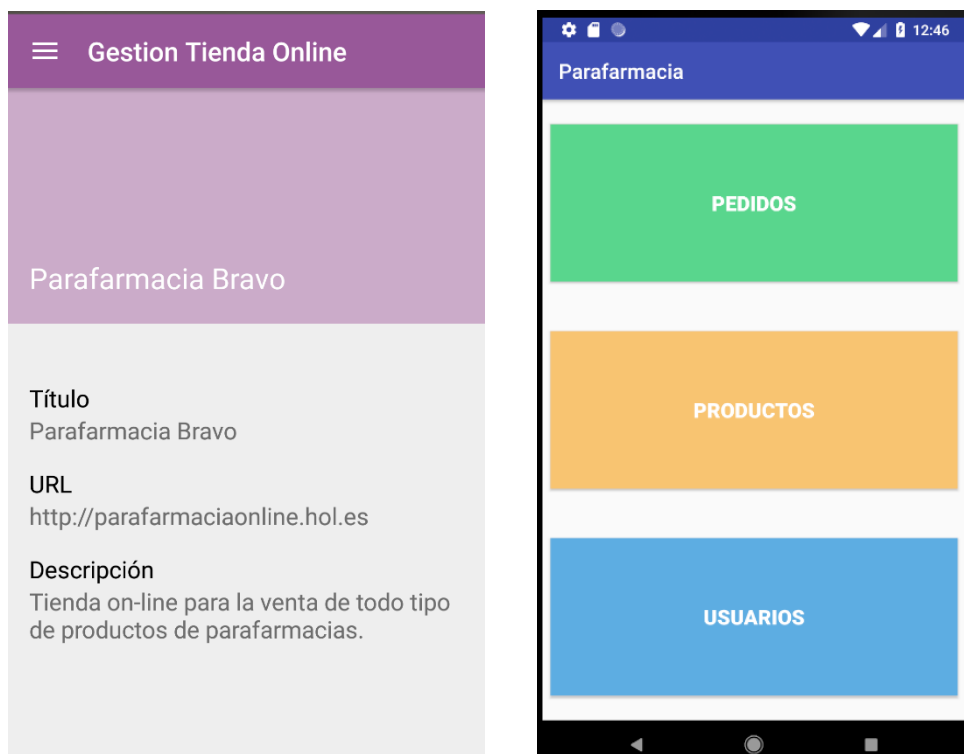


Ilustración 135 Pantalla principal cliente Android TFG y TFM

Pedidos

Para las pantallas de pedidos, tenemos una clara diferencia, en la versión antigua, tenemos todos los textos mezclados, con diferentes tipos, tamaños, sin apenas orden. Para la nueva versión, se han cuidado esos pequeños detalles, en primer lugar, nos podemos fijar que el color verde del botón de pedidos en la pantalla principal se mantiene, además tenemos dos apartados claramente definidos, productos y datos de envío. Por otro lado, se han incluido tres botones con las opciones básicas que podemos hacer con el pedido, los colores utilizados para cada opción se han sacado de una guía que identifica los estados de pedidos por colores, para intentar tenerlo lo más estandarizado posible.



Ilustración 136 Pantalla pedidos cliente Android TFG y TFM

Productos

La lista de productos es similar para ambas versiones, en la nueva versión se ha organizado para que el nombre del producto tenga más espacio a lo ancho y sea más fácil de leer, además, se han incluido botones con las opciones que podemos realizar sobre los productos. La versión antigua se accede si pinchamos encima de uno de ellos, pero se ha creído que es más accesible y usable la nueva versión, ya que indicamos claramente las funciones que tenemos disponibles. Por otro lado, se ha cambiado el botón para crear un nuevo producto, ajustándose el nuevo a los estándares de altura y tamaño de texto, así como introduciendo el texto descriptivo.

Como última diferencia, las imágenes de la nueva versión se van cargando según nos desplazamos por la lista de productos hacia abajo para que cargue más rápido mientras en la antigua se cargaban todos a la vez.

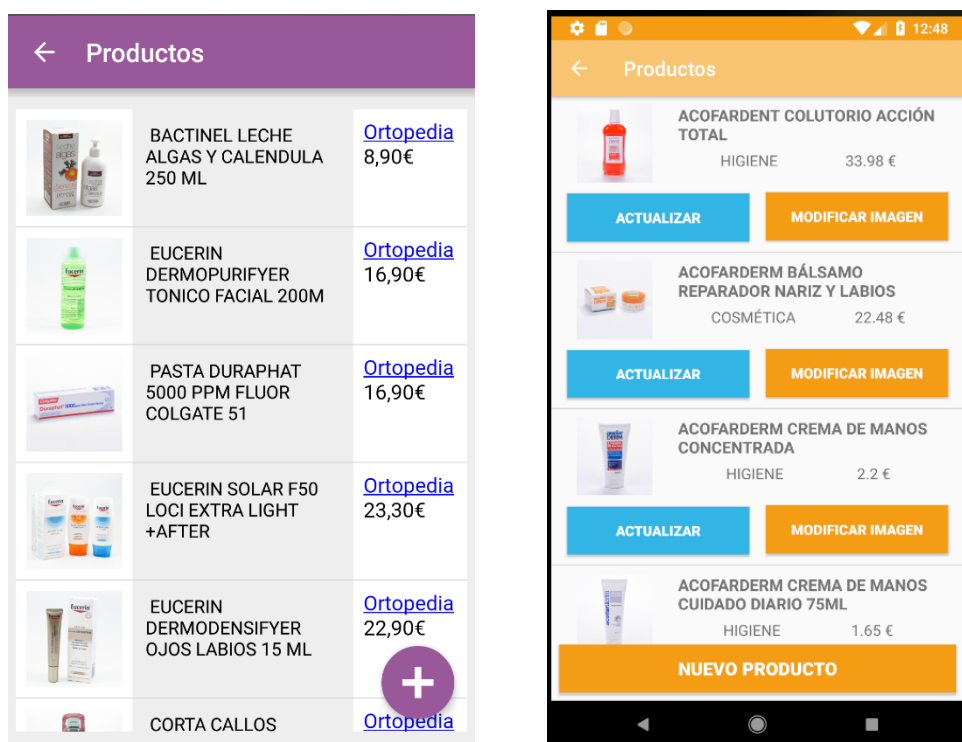


Ilustración 137 Pantalla productos cliente Android TFG y TFM

10.4 Generador

En lugar de realizar una comparación entre el generador antiguo y el nuevo, se va a realizar una lista de las novedades que incluye la nueva versión, ya que la antigua únicamente generaba un archivo con el script para la base de datos de Wordpress.

1. **Validaciones:** se ha incluido un validador al generador que permite comprobar ciertas partes del lenguaje que podrían causar algún conflicto en caso de no ser correctos, por ejemplo, los códigos de colores.
2. **Formateo:** se ha añadido la opción de formatear el código del lenguaje para hacerlo más limpio de cara al usuario.
3. **Templates:** se han añadido ayudas para escribir el código de generación mediante Templates. Estas ayudas aparecen cuando comenzamos a escribir el código desde cero o cuando creamos un nuevo producto.
4. **Generador JSP:** se ha añadido un generador que permite crear páginas JSP con las vistas de la parafarmacia.
5. **Generador CSS:** se ha añadido un generador que permite crear la hoja de estilos utilizada por la parafarmacia.
6. **Generador Base de datos:** se ha añadido un generador que se conecta a los servicios web para limpiar la base de datos e insertar nuevos productos.

Capítulo 11. Conclusiones y Ampliaciones

11.1 Conclusiones

Una vez finalizado el desarrollo del proyecto, se ha conseguido cumplir con todos los objetivos marcados al inicio del mismo, la parafarmacia web es más completa de lo que se había planificado en un principio, ya que se han ido añadiendo pequeños detalles que no se habían tenido en cuenta, obtenidos de la experiencia de trabajar en una empresa de desarrollo web.

El cliente Android se ha integrado a la perfección con la parafarmacia y se han añadido mejoras de rendimiento para conseguir una aplicación usable, sin grandes tiempos de espera, mediante el uso de la caché entre otras cosas.

El generador también ha superado las expectativas, ya que se han añadido algunos detalles que lo hacen más completo y fácil de usar y no se había tenido en cuenta al iniciar el proyecto, por ejemplo, el uso de los servicios web para inicializar los productos, la compilación y despliegue automático del código generado o el uso de Templates para escribir el código.

En conjunto, se puede afirmar que el producto desarrollado es de gran calidad y supera por mucho al desarrollado hace unos años como Trabajo Final de Grado, la elección de las tecnologías ha sido un acierto, ya que se complementan mejor y su uso conjunto tiene más sentido que la combinación escogida para el antiguo trabajo.

Para concluir, hacer mención al estudio previo de las diferentes tecnologías utilizadas y aprendidas durante el Máster, junto con la planificación de la arquitectura general, que ha sido clave para poder desarrollar un sistema tan completo y para evitar que surgieran grandes problemas o incompatibilidades durante la fase de desarrollo, como ocurrió con el Trabajo Fin de Grado.

11.2 Ampliaciones

A continuación, se van a describir las posibles mejoras que se pretenden introducir en el sistema para el futuro.

11.2.1 Métodos de pago

En un principio, la parafarmacia web cuenta con las opciones de pago más básicas, contra-rembolso y Paypal con sus diferentes opciones de pagos con tarjeta, cuando se ponga el sistema en producción se valorará la opción de incluir nuevos métodos de pago que permitan al usuario tener más posibilidades a la hora de elegir, además, junto con esta ampliación, se incluirá en el generador estas nuevas opciones para hacerlo aún más completo, ya que permitirá crear una tienda con más posibilidades y adecuarla a las necesidades del cliente.

11.2.2 Ofertas para productos

Actualmente, la parafarmacia web tiene la opción de incluir los productos que queramos, pero no se puede poner carteles de ofertas o precios tachados con el descuento realizado, para llamar la atención del cliente, por esta razón, una de las ampliaciones a realizar cuando se ponga en marcha la tienda, será incluir la opción de productos en oferta.

Esta mejora no se ha incluido ya que no estaba entre los requisitos iniciales, pero se ha pensado que puede ser una opción muy positiva en la tienda a la hora de vender.

11.2.3 Recuperar contraseñas

Los usuarios que se registran en la tienda deben introducir como información para iniciar sesión el usuario y la contraseña, pero en caso de olvidarse de la contraseña, la única opción que disponen es la de ponerse en contacto con el administrador de la tienda para que restablezca la contraseña. Para mejorar esta solución, se incluirá la opción de restablecer la contraseña enviando un correo electrónico con una URL destinada a tal efecto. Esto no se ha incluido en esta versión por falta de tiempo, pero se piensa añadir antes de hacer pública la tienda.

11.2.4 SEO

Hoy en día, una de las partes más importantes para vender por Internet es tener una buena posición en los buscadores, esto se consigue con una serie de trucos como la elección de palabras clave, o teniendo un contenido único y escrito correctamente entre otras muchas cosas. Ya que en el Máster no tenemos ninguna asignatura que se centre en este tema y además, puede llevar mucho tiempo preparar la tienda para ello, se ha decidido dejar como mejora futura que se irá implementando poco a poco cuando se comience a vender.

Capítulo 12. Referencias Bibliográficas

1. **Institute, Project Management.** *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK) Quinta edición.* Pensilvania : FSC.
2. **Design, Material.** Material Design. [En línea] 2018. <https://material.io/design/usability/accessibility.html#understanding-accessibility>.
3. **PrestaShop.** PrestaShop. [En línea] 8 de 10 de 2017. <http://doc.prestashop.com/display/PS16/Estados+de+Pedidos>.
4. **Gijón, Centro Tecnológico de.** W3C España. [En línea] 6 de 6 de 2018. <https://www.w3c.es/>.
5. **Flower.** [En línea] 2 de 7 de 2010. <http://javafoundations.blogspot.com.es/2010/07/java-estandares-de-programacion.html>.
6. **España, Gobierno de.** [En línea] <https://www.boe.es/buscar/doc.php?id=BOE-A-1999-23750>.
7. —. [En línea] <https://www.boe.es/buscar/act.php?id=BOE-A-2002-13758>.
8. **Montero, Yusef Hassan.** Guía de Evaluación Heurística de Sitios Web. [En línea] 2003. <http://www.nosolousabilidad.com/articulos/heuristica.htm>.
9. **Google.** [En línea] https://search.google.com/test/mobile-friendly?utm_source=mft&utm_medium=redirect&utm_campaign=mft-redirect&hl=ES.
10. **Schaeffer, Kyle.** [En línea] <http://quirktools.com/screenfly/>.
11. **Alicante, Universidad de.** Accesibilidad Web. [En línea] 2018. <http://accesibilidadweb.dlsi.ua.es/?menu=puntos-1.0>.
12. **Exes.** XML ¿Qué es? [En línea] <http://www.mundolinux.info/que-es-xml.htm>.
13. **IBM.** ¿Qué es WSDL? [En línea] https://www.ibm.com/support/knowledgecenter/es/SSMKHH_9.0.0/com.ibm.etools.mft.doc/ac34640_.htm.
14. —. UDDI. [En línea] https://www.ibm.com/support/knowledgecenter/es/SS4JE2_7.5.5/org.eclipse.jst.ws.consumption.ui.doc.user/concepts/cuddi.html.
15. **Wikipedia.** WS-Security. [En línea] 3 de 2013. <https://es.wikipedia.org/wiki/WS-Security>.
16. **Marqués, Asier.** Conceptos sobre APIs REST. [En línea] 11 de 4 de 2013. <http://asiermarques.com/2013/conceptos-sobre-apis-rest/>.

17. **Foundation, Eclipse.** Xtext. [En línea] <https://www.eclipse.org/Xtext/>.
18. **Desarrolladores.** Stackoverflow. [En línea] 2008. <https://stackoverflow.com/>.
19. **Paypal.** Guía de integración de Pasarela integral. [En línea] 2016. https://www.paypalobjects.com/webstatic/es_ES/developer/docs/pdf/pasarelaintegral_es.pdf.
20. **Square.** Picasso. [En línea] <http://square.github.io/picasso/>.
21. **Freepik.** Recursos gráficos para todos. [En línea] <https://www.freepik.es/>.
22. **Adobe.** Rueda cromática. [En línea] <https://color.adobe.com/es/create/color-wheel/>.
23. **Codepen.** Social development environment. [En línea] <https://codepen.io/>.
24. **Amodeo, Enrique.** ¿Qué son los DSL? [En línea] 2010. <https://eamodeorubio.wordpress.com/2010/09/13/%C2%BFque-son-los-dsl-domain-specific-languages/>.
25. **RubenFa.** MongoDB. [En línea] 3 de 2 de 2014. <https://www.genbetadev.com/bases-de-datos/mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no>.
26. **Wikipedia.** SQLite. [En línea] 9 de 5 de 2018. <https://es.wikipedia.org/wiki/SQLite>.
27. **Spring.** Spring. [En línea] 8 de 5 de 2018. <https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html>.
28. **Yanira.** Política de cookies. [En línea] <https://blog.ensalza.com/politica-de-cookies/>.

Capítulo 13. Apéndices

13.1 Glosario y Diccionario de Datos

Por orden alfabético, todos los términos que se consideren importantes en la aplicación con una descripción breve de su significado dentro de la aplicación.

- **Activity**: son las clases de una aplicación Android encargadas de crear las pantallas.
- **APK**: es la extensión del archivo que permite instalar una aplicación Android en un dispositivo.
- **CSS**: Cascading Style Sheets, son las hojas de estilo que permiten definir la presentación de un documento estructurado.
- **DAO**: Data Access Object, es un patrón de arquitectura enfocado en proveer una interfaz común para acceder a múltiples dispositivos de almacenamiento.
- **DSL**: Domain Specific Language, es un lenguaje dedicado a resolver un problema específico.
- **HTML**: HyperText Markup Language, es el lenguaje de marcado para crear páginas web.
- **IBM**: International Business Machines Corporation, es una empresa multinacional de tecnología.
- **JSP**: JavaServer Pages, es una tecnología que permite crear aplicaciones web dinámicas basadas principalmente en HTML.
- **LSSI**: Ley de servicios de la sociedad de la información.
- **MVC**: Modelo Vista Controlador, es un patrón de arquitectura software utilizado comúnmente en aplicaciones web, permite separar los datos, la lógica y la interfaz de usuario.
- **SOAP**: Simple Object Access Protocol, es un protocolo estándar para definir cómo se van a comunicar dos objetos mediante el uso de XML.
- **TAW**: Test de Accesibilidad Web.
- **TFG**: Trabajo Fin de Grado.
- **TFM**: Trabajo Fin de Máster.
- **W3C**: World Wide Web Consortium, es un consorcio que genera estándares para asegurar el crecimiento de la web.
- **WCAG**: Web Content Accessibility Guidelines, son un conjunto de pautas de accesibilidad para la web, desarrollado por el W3C.
- **XML**: eXtensible Markup Language, es un meta-lenguaje para definir lenguajes de marcado.

13.2 Contenido Entregado Digitalmente

13.2.1 Contenidos

Directorio	Contenido
<i>./ Directorio raíz</i>	Contiene todos los ficheros que componen el proyecto.
<i>./Cliente Android</i>	Contiene todos los archivos relacionados con el cliente Android.
<i>./Cliente Android/Apk</i>	Contiene el archivo <i>.apk</i> necesario para realizar la instalación de la aplicación en un dispositivo con sistema operativo Android.
<i>./Cliente Android/GestorParafarmacia</i>	Directorio que contiene el proyecto del cliente Android.
<i>./Cliente Android/ GestorParafarmacia /app</i>	Contiene las librerías del proyecto y la carpeta <i>src</i> .
<i>./Cliente Android/GestorParafarmacia /app/src</i>	Contiene los ficheros con las pruebas de la aplicación y el código fuente.
<i>./Documentacion</i>	Contiene la documentación del proyecto y otro directorio con imágenes.
<i>./Documentacion/imagenes</i>	Contiene los directorios con los diagramas del sistema, los logos y los mapas de pantalla.
<i>./Documentacion/imágenes/diagramas</i>	Contiene las imágenes con los diagramas del sistema.
<i>./Documentacion/imágenes/logos</i>	Contiene las imágenes de los distintos logos usados en los subsistemas.
<i>./Documentacion/imágenes/mapa de pantallas</i>	Contiene las imágenes con los mapas de pantallas usados en los subsistemas.
<i>./Generador de parafarmacias</i>	Contiene todos los archivos relacionados con el generador de parafarmacias.
<i>./Generador de parafarmacias/Generador</i>	Contiene el proyecto eclipse con los archivos y el código desarrollado.
<i>./Generador de parafarmacias/EclipseApplication</i>	Contiene el proyecto encargado de personalizar la parafarmacia generada. Se

	lanza a partir del proyecto Generador.
./Parafarmacia Web	Contiene los archivos relacionados con la parafarmacia web.
./Parafarmacia Web/Base de datos	Contiene el script <i>.sql</i> con la base de datos de la parafarmacia web.
./Parafarmacia Web/Parafarmacia	Contiene todos los directorios y ficheros creados para el desarrollo de la parafarmacia web.
./Parafarmacia Web/Subir imagen	Contiene todos los ficheros PHP encargados de subir imágenes a la parafarmacia.
./Servicios Web	Contiene todos los ficheros creados para el desarrollo de los servicios web.

Tabla 56 Contenido entregado

13.3 Índice Alfabético

A

accesibilidad, 7, 14, 40, 66, 67, 99, 134, 141, 151, 160, 167, 170, 214, 229
 Android, 6, 30, 62, 64, 99, 107, 179, 180, 229, 230
 Android Studio, 107

B

base de datos, 59, 60, 231
 bases de datos, 26, 27, 29, 109, 110, 113, 201

C

cliente Android, 11, 12, 13, 14, 18, 19, 22, 26, 30, 31, 37, 42, 43, 49, 50, 53, 54, 58, 59, 61, 62, 68, 69, 75, 76, 78, 83, 92, 93, 94, 95, 96, 97, 101, 104, 107, 110, 112, 121, 122, 123, 124, 125, 130, 131, 174, 179, 191, 192, 193, 194, 195, 196, 197, 198, 199, 202, 217, 218, 219, 220, 222
 Cliente Android, 230
 CSS, 103

D

DSL, 6, 7, 12, 25, 29, 60, 77, 111, 228, 229

E

Eclipse, 8, 11, 12, 25, 55, 101, 105, 107, 111, 180, 181, 227
 estándar, 26, 30, 99, 102, 104, 138, 140, 205, 229

G

generador, 1, 5, 6, 11, 12, 14, 17, 18, 19, 22, 25, 29, 30, 31, 37, 41, 48, 49, 51, 52, 53, 55, 56, 60, 61, 62, 65, 68, 69, 77, 78, 98, 101, 105, 111, 113, 126, 128, 141, 148, 174, 180, 181, 203, 205, 214, 217, 221, 222, 223
 Generador, 230, 231

H

HTML, 102

I

Illustrator, 108

interfaz, 11, 19, 30, 49, 51, 52, 54, 55, 56, 72, 104, 139, 141, 169, 218, 229

J

Java, 23, 26, 99, 101, 102

L

Las pruebas de usabilidad, 62
 lenguaje de dominio específico, 6, 29
 Ley, 8, 39, 99, 100, 229

P

parafarmacia web, 11, 12, 13, 14, 19, 22, 24, 30, 31, 37, 38, 40, 42, 43, 49, 50, 51, 52, 53, 54, 57, 59, 61, 62, 64, 68, 69, 71, 77, 78, 81, 83, 84, 85, 86, 87, 88, 89, 90, 91, 99, 100, 101, 102, 103, 106, 109, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 125, 128, 130, 142, 150, 157, 164, 174, 177, 178, 182, 183, 184, 185, 186, 187, 188, 189, 190, 192, 201, 215, 216, 217, 222, 223
 Parafarmacia Web, 231
 PHP, 6

S

servicio web, 14, 30, 51, 77, 121, 128, 131, 202, 217, 218
 SOAP, 7, 30, 49, 51, 52, 69, 121, 202, 217, 229
 Spring MVC, 23, 101, 205
 SQL, 29

U

usabilidad, 7, 14, 24, 62, 64, 66, 137

W

Webservice, 30
 Wordpress, 231

X

Xtext, 12, 25, 101, 105, 111, 180, 181, 227

