

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UN SISTEMA WEB DE COMERCIO ELECTRÓNICO B2C, PARA LA PROMOCIÓN, COMPRA ON-LINE Y GESTIÓN DE STOCK DE ARTÍCULOS DE CUERO

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

RENÁN MAURICIO GUERRERO CANDO
renan_mauricio@hotmail.com

MARÍA FERNANDA GUERRERO HERRERA
ferpgn@gmail.com

DIRECTOR: MSC ING. VILLAVICENCIO OTÁÑEZ FRANCISCO HORACIO
fvillavi@yahoo.com

Quito, Diciembre 2014

DECLARACIÓN

Nosotros, Renán Mauricio Guerrero Cando y María Fernanda Guerrero Herrera declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Renán Guerrero

María Fernanda Guerrero

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Renán Mauricio Guerrero Cando y María Fernanda Guerrero Herrera, bajo mi supervisión.

Francisco Villavicencio
DIRECTOR DE PROYECTO

AGRADECIMIENTO

A Dios, por darme la fortaleza para seguir adelante a pesar de los tropiezos que se presentan en la vida.

A mis padres, Carmen Cando y Luis Guerrero, que de forma incondicional siempre han estado pendientes ante mis victorias y derrotas, y que sobre todo en esos momentos grises, han sido mi soporte y motor para levantarme y seguir luchando por mis sueños.

A Fernanda, que además de ser mi compañera, es mi amiga, y que sin su valiosa ayuda no hubiera sido posible culminar este objetivo.

Al Ing. Francisco Villavicencio, por su paciencia y guía en todo momento para que el fruto de nuestro trabajo se vea recompensado.

A mi familia y amigos, que con sus palabras me han brindado ánimo para alcanzar esta meta.

Renán Guerrero.

AGRADECIMIENTO

A mi familia, por estar a mi lado apoyándome en todas las batallas que he tenido que afrontar.

A mis amigos por hacerme sonreír y ayudarme a mi mente disipar.

A mis hijos por existir.

A mi compañero de tesis, Renán, que más que compañero es mi amigo incondicional, gracias por emprender conmigo esta travesía y estar resuelto a llegar hasta el final.

A mi tutor y más querido profesor, al Ingeniero Francisco Villavicencio, gracias por su guía, su paciencia y por todos los conocimientos que nos transmitió. Su ayuda fue valiosa en todo momento.

Fernanda

DEDICATORIA

A mi madre, pilar fundamental de mi vida, quien a pesar de la distancia, su cariño y apoyo lograron traspasar fronteras, y sé que la culminación de esta meta, es fruto de ese amor incondicional.

A mi padre, ejemplo a seguir, quien con sus consejos y apoyo, han dejado como legado la constancia para siempre perseguir y alcanzar mis sueños.

A Fernanda, mi buena amiga, con quien aún con los traspies que se presentaron, logramos alcanzar nuestro objetivo.

A todos aquellos que creyeron en mí, y que se hicieron partícipes de este logro.

Renán Guerrero.

DEDICATORIA

A la memoria del hombre que más ha influido en mi vida, mi mayor ejemplo de trabajo, a mi padre, sus palabras de aliento siguen perennes en mi mente y me han impulsado a alcanzar mis metas pese a todos los traspies que el camino ha presentado.

A mis tres hijos. El motor que mueve mi mundo. A Giuliana por sus cartitas, a Nicole por las risas incontroladas, a Thiago por la ternura en su mirar. A ustedes 3 hijos míos gracias por darle luz a mi vida y con solo una sonrisa iluminar mis momentos de oscuridad.

A mi madre, por su ayuda y paciencia, sin su ayuda madre, no habría conseguido levantarme las veces que me vi caer.

Fernanda

CONTENIDO

CONTENIDO.....	viii
RESUMEN	xv
PRESENTACIÓN	xvi
1 CAPÍTULO 1. PLANTEAMIENTO DEL PROBLEMA.....	1
1.1 DEFINICIÓN DEL PROBLEMA.....	1
1.1.1 ANTECEDENTES DEL PROBLEMA.....	1
1.1.2 COMERCIO ELECTRÓNICO	2
1.1.3 JUSTIFICACIÓN DEL MODELO DE COMERCIO ELECTRÓNICO APLICADO AL PROYECTO	7
1.1.4 FACTIBILIDAD DEL PROYECTO	7
1.1.5 BENEFICIOS DEL PROYECTO	10
1.2 SELECCIÓN DE LA METODOLOGÍA DE DESARROLLO	11
1.2.1 METODOLOGÍAS TRADICIONALES.....	11
1.2.2 METODOLOGÍAS ÁGILES.....	13
1.2.3 COMPARACIÓN ENTRE METODOLOGÍAS DE DESARROLLO	14
1.2.4 DESCRIPCIÓN DE LA METODOLOGÍA ESCOGIDA (HIBRIDA: SCRUM Y XP)	16
1.3 JUSTIFICACIÓN DE LAS HERRAMIENTAS DE DESARROLLO.....	24
1.3.1 SOFTWARE LIBRE	24
1.3.2 GESTORES DE BASES DE DATOS.....	25
1.3.3 LENGUAJE PHP	26
1.3.4 SERVIDOR WEB APACHE	27
1.3.5 SCRUMY: HERRAMIENTA PARA GESTIONAR SCRUM	28
1.3.6 APTANA: IDE PARA DESARROLLO EN PHP	28
2 CAPÍTULO 2. DESARROLLO DEL SISTEMA.....	29
2.1 ESPECIFICACIÓN DE REQUERIMIENTOS	29
2.1.1 USUARIOS DEL SISTEMA	29
2.1.2 HISTORIAS DE USUARIO	30
2.1.3 DEFINICION DE LAS HISTORIAS DE USUARIO.....	31
2.2 ANÁLISIS	45

2.2.1	PRODUCT BACKLOG.....	45
2.2.2	SPRINT BACKLOG Y BURN DOWN CHART	47
2.2.3	PLANIFICACIÓN INICIAL DE LAS ITERACIONES/SPRINTS	52
2.3	<i>DISEÑO</i>	52
2.3.1	DESCRIPCIÓN DEL PATRÓN DE ARQUITECTURA.....	52
2.3.2	MODELO DE BASE DE DATOS	54
2.3.3	MODELO DE NAVEGABILIDAD	54
2.4	IMPLEMENTACIÓN	57
2.4.1	ESTÁNDAR DE CODIFICACIÓN	57
2.4.2	IMPLEMENTACIÓN ACORDE AL PATRÓN MVC (MODELO VISTA CONTROLADOR).....	58
2.4.3	EJECUCIÓN REAL DE LAS ITERACIONES/SPRINTS	62
2.4.4	COMPARACIÓN DE LA PLANIFICACIÓN INICIAL DE LAS ITERACIONES/SPRINTS VS LA IMPLEMENTACIÓN REAL	92
2.4.5	EJEMPLOS DEL CÓDIGO FUENTE EN BASE AL MODELO MVC. 92	
2.5	PRUEBAS	101
2.5.1	PRUEBAS UNITARIAS	102
3	CAPÍTULO 3. IMPLANTACIÓN Y ANÁLISIS DE RESULTADOS	108
3.1	ANÁLISIS DE INSTRUMENTOS DE IMPLANTACIÓN.....	108
3.1.1	SOFTWARE RECOMENDADO.....	108
3.1.2	HARDWARE RECOMENDADO	108
3.1.3	ALOJAMIENTO VPS (VIRUAL SERVER)	109
3.2	PRUEBAS	109
3.2.1	PRUEBAS DE ACEPTACIÓN	110
3.2.2	PRUEBAS DE COMPATIBILIDAD DEL SISTEMA CON LOS EXPLORADORES MÁS USADOS	124
3.2.3	PRUEBAS DE RENDIMIENTO.....	126
3.3	ACEPTACIÓN.....	127
3.4	ANÁLISIS DE RESULTADOS.....	128
3.4.1	RESULTADOS DE LAS PRUEBAS DE ACEPTACIÓN (FUNCIONALIDAD).....	128
3.4.2	RESULTADOS DE LA ENCUESTA A LOS USUARIOS FINALES. 130	
4	CONCLUSIONES Y RECOMENDACIONES.....	134
4.1	CONCLUSIONES.....	134

4.2 RECOMENDACIONES	135
GLOSARIO.....	138
BIBLIOGRAFIA	139
LIBROS – REVISTAS	139
DIRECCIONES ELECTRONICAS	143
ANEXOS	147
ACTA DE REUNIÓN DAILY MEETING.....	148
ACTA DE REUNIÓN SPRINT PLANNING MEETING.....	149
ACTA DE REUNIÓN PRODUCT BACKLOG REFINEMENT	150
ACTA DE REUNIÓN SPRINT RETROSPECTIVE	151
ACTA DE REUNIÓN SPRINT REVIEW	152

INDICE DE TABLAS

TABLA 1.1: :COMPARACIÓN ENTRE DIFERENTES SERVICIOS DE HOSTING VPS.....	10
TABLA 1.2: COMPARACIÓN ENTRE GRUPOS DE METODOLOGÍAS DE DESARROLLO	15
TABLA 2.1: FORMATO DE HISTORIA DE USUARIO	30
TABLA 2.2: HISTORIA DE USUARIO INGRESAR NUEVO USUARIO.....	31
TABLA 2.3: HISTORIA DE USUARIO MODIFICAR USUARIO	31
TABLA 2.4: HISTORIA DE USUARIO ELIMINAR USUARIO.....	32
TABLA 2.5: HISTORIA DE USUARIO IMPLEMENTACIÓN DE PERFILES	32
TABLA 2.6: HISTORIA DE USUARIO INGRESAR NUEVAS CATEGORÍAS.....	33
TABLA 2.7: HISTORIA DE USUARIO MODIFICAR CATEGORÍAS	33
TABLA 2.8: HISTORIA DE USUARIO ELIMINAR CATEGORÍAS.....	34
TABLA 2.9: HISTORIA DE USUARIO INGRESAR NUEVOS PRODUCTOS.....	34
TABLA 2.10: HISTORIA DE USUARIO MODIFICAR PRODUCTOS.....	35
TABLA 2.11: HISTORIA DE USUARIO ELIMINAR PRODUCTOS	35
TABLA 2.12: HISTORIA DE USUARIO INGRESAR EMPRESAS PROVEEDORAS.....	36
TABLA 2.13: HISTORIA DE USUARIO MODIFICAR EMPRESAS PROVEEDORAS	36
TABLA 2.14: HISTORIA DE USUARIO ELIMINAR EMPRESAS PROVEEDORAS.....	37
TABLA 2.15: ESTABLECER FORMAS DE PAGO.....	37
TABLA 2.16: HISTORIA DE USUARIO GESTIÓN DE PEDIDOS, COMPRAS Y PAGOS	38
TABLA 2.17: HISTORIA DE USUARIO MANEJO DE STOCK	38
TABLA 2.18: HISTORIA DE USUARIO INFORMACION DE VENTAS.....	39
TABLA 2.19: HISTORIA DE USUARIO INGRESAR INFORMACIÓN DE LA EMPRESA	39
TABLA 2.20: HISTORIA DE USUARIO MODIFICAR INFORMACIÓN DE LA EMPRESA	40
TABLA 2.21: HISTORIA DE USUARIO ELIMINAR INFORMACIÓN DE LA EMPRESA.....	40
TABLA 2.22: HISTORIA DE USUARIO CREAR CUENTA DE USUARIO	41
TABLA 2.23: HISTORIA DE USUARIO MODIFICAR CUENTA DE USUARIO.....	41
TABLA 2.24: HISTORIA DE USUARIO ELIMINAR CUENTA DE USUARIO	42
TABLA 2.25: HISTORIA DE USUARIO BUSCAR UN PRODUCTO	42
TABLA 2.26: HISTORIA DE USUARIO INGRESAR NUEVO USUARIO	43
TABLA 2.27: HISTORIA DE USUARIO ELIMINAR UN PRODUCTO ANTES DE REALIZAR LA COMPRA	43
TABLA 2.28: HISTORIA DE USUARIO REALIZAR UNA COMPRA.....	44
TABLA 2.29: FORMATO DEL PRODUCT BACKLOG	46
TABLA 2.30: PRODUCT BACKLOG	49
TABLA 2.31: FORMATO DEL SPRINT BAKLOG.....	51
TABLA 2.32: DIVISIÓN IDEAL DEL PRODUCT BACKLOG EN SPRINTS (ITERACIONES).....	53
TABLA 2.33: SPRINT BACKLOG 1	64
TABLA 2.34: SPRINT BACKLOG 2	69
TABLA 2.35: SPRINT BACKLOG 3	75
TABLA 2.36: SPRINT BACKLOG 4	79
TABLA 2.37: SPRINT BACKLOG 5	82
TABLA 2.38: SPRINT BACKLOG 6	85
TABLA 2.39: SPRINT BACKLOG 7	90
TABLA 2.40: SPRINTS PLANIFICADOS VS SPRINTS IMPLEMENTADOS.....	93
TABLA 2.41: ASERCIONES USADAS CON MÁS FRECUENCIA EN LAS PRUEBAS PHPUNIT	107
TABLA 3.1: ESPECIFICACIÓN DEL SOFTWARE REQUERIDO	108
TABLA 3.2: ESPECIFICACIÓN DEL HARDWARE REQUERIDO	109

TABLA 3.3: PLANTILLA PRUEBAS DE ACEPTACIÓN	110
TABLA 3.4: PRUEBA DE ACEPTACIÓN – INGRESAR NUEVO USUARIO	111
TABLA 3.5: PRUEBA DE ACEPTACIÓN – MODIFICAR USUARIO EXISTENTE	112
TABLA 3.6: PRUEBA DE ACEPTACIÓN – ELIMINAR USUARIO.....	113
TABLA 3.7: PRUEBA DE ACEPTACIÓN – INGRESAR NUEVAS CATEGORÍAS	113
TABLA 3.8: PRUEBA DE ACEPTACIÓN – MODIFICAR CATEGORÍAS	114
TABLA 3.9: PRUEBA DE ACEPTACIÓN – ELIMINAR CATEGORÍAS	114
TABLA 3.10: PRUEBA DE ACEPTACIÓN – INGRESAR NUEVOS PRODUCTOS	115
TABLA 3.11: PRUEBA DE ACEPTACIÓN – MODIFICAR PRODUCTOS	115
TABLA 3.12: PRUEBA DE ACEPTACIÓN – ELIMINAR PRODUCTOS	116
TABLA 3.13: PRUEBA DE ACEPTACIÓN – INGRESAR NUEVOS FABRICANTES.....	116
TABLA 3.14: PRUEBA DE ACEPTACIÓN – MODIFICAR FABRICANTES.....	117
TABLA 3.15: PRUEBA DE ACEPTACIÓN – ELIMINAR FABRICANTES	117
TABLA 3.16: PRUEBA DE ACEPTACIÓN – INFORMACIÓN DE VENTAS	118
TABLA 3.17: PRUEBA DE ACEPTACIÓN – INGRESAR INFORMACIÓN DE LA EMPRESA	119
TABLA 3.18: PRUEBA DE ACEPTACIÓN – MODIFICAR INFORMACIÓN DE LA EMPRESA.....	119
TABLA 3.19: PRUEBA DE ACEPTACIÓN – ELIMINAR INFORMACIÓN DE LA EMPRESA	120
TABLA 3.20: PRUEBA DE ACEPTACIÓN – CREAR CUENTA DE USUARIO	120
TABLA 3.21: PRUEBA DE ACEPTACIÓN – MODIFICAR CUENTA DE USUARIO	121
TABLA 3.22: PRUEBA DE ACEPTACIÓN – ELIMINAR CUENTA DE USUARIO	121
TABLA 3.23: PRUEBA DE ACEPTACIÓN – BUSCAR UN PRODUCTO	122
TABLA 3.24: PRUEBA DE ACEPTACIÓN – AÑADIR UN PRODUCTO AL CARRO DE COMPRA	122
TABLA 3.25: PRUEBA DE ACEPTACIÓN – ELIMINAR UN PRODUCTO ANTES DE REALIZAR UNA COMPRA	123
TABLA 3.26: PRUEBA DE ACEPTACIÓN – REALIZAR UNA COMPRA.....	123
TABLA 3.27: EVALUACIÓN DE LAS CARACTERÍSTICAS DEL SISTEMA VISTAS EN LOS DIFERENTES NAVEGADORES	126
TABLA 3.28: RESUMEN DE LOS RESULTADOS DE LAS PRUEBAS DE ACEPTACIÓN	129
TABLA 3.29: RESPUESTAS A LA PREGUNTA 1.....	130
TABLA 3.30: RESPUESTAS A LA PREGUNTA 2.....	131
TABLA 3.31: RESPUESTAS A LA PREGUNTA 3.....	131
TABLA 3.32: RESPUESTAS A LA PREGUNTA 4.....	132
TABLA 3.33: RESPUESTAS A LA PREGUNTA 5.....	133

INDICE DE GRAFICOS

FIGURA 1.1: PROCESO DE SCRUM	19
FIGURA 1.2: PROCESO SCRUM CON XP	22
FIGURA 2.1: SCRUM TASK BOARD REALIZADO USANDO LA HERRAMIENTA EN LÍNEA SCRUMY.....	50
FIGURA 2.2: FUNCIONAMIENTO DEL PATRÓN DE DISEÑO MODELO-VISTA-CONTROLADOR.....	54
FIGURA 2.3: DIAGRAMA DE NAVEGABILIDAD EN EL BACKEND	55
FIGURA 2.4: . DIAGRAMA DE NAVEGABILIDAD PARA EL USUARIO NO REGISTRADO	55
FIGURA 2.5: DIAGRAMA DE NAVEGABILIDAD PARA EL USUARIO REGISTRADO	55
FIGURA 2.6: MODELO ENTIDAD RELACIÓN DEL SISTEMA ESIWECOM	57
FIGURA 2.7: VISIÓN GENERAL DE LOS ELEMENTOS DEL SISTEMA EN APTANA	59
FIGURA 2.8: VISIÓN DE LOS DIRECTORIOS QUE FORMAN EL CORE DEL SISTEMA.....	61
FIGURA 2.9: BURN DOWN CHART SPRINT 1	65
FIGURA 2.10: BUSCAR PRODUCTO	66
FIGURA 2.11: BÚSQUEDA AVANZADA	66
FIGURA 2.12: AÑADIR PRODUCTO AL CARRITO DE COMPRA	67
FIGURA 2.13: ELIMINAR PRODUCTO DE LA COMPRA	67
FIGURA 2.14: CREAR CUENTA DE USUARIO	68
FIGURA 2.15: INTERFAZ RESULTANTE DE LA PRIMERA ITERACIÓN (SPRINT 1).....	68
FIGURA 2.16: BURN DOWN CHART SPRINT 2	71
FIGURA 2.17: CREAR CUENTA DE USUARIO (FRONT END)	71
FIGURA 2.18: CREAR CUENTA DE USUARIO (BACK END)	72
FIGURA 2.19: REALIZAR LA COMPRA – INFORMACIÓN SOBRE EL ENVÍO	72
FIGURA 2.20: REALIZAR LA COMPRA – INFORMACIÓN SOBRE EL PAGO.....	73
FIGURA 2.21: REALIZAR LA COMPRA – CONFIRMACIÓN.....	73
FIGURA 2.22: MODIFICAR CUENTA DE USUARIO (FRONT END).....	74
FIGURA 2.23: ELIMINAR CUENTA DE USUARIO (EN EL BACKEND).....	74
FIGURA 2.24: BURN DOWN CHART SPRINT 3	76
FIGURA 2.25: IMPLEMENTACIÓN DE GRUPOS DE USUARIOS (BACK END)	77
FIGURA 2.26: CREAR USUARIO.....	78
FIGURA 2.27: INTERFAZ DE LOGIN.....	78
FIGURA 2.28: BURN DOWN CHART SPRINT 4	80
FIGURA 2.29: INGRESAR CATEGORÍAS	80
FIGURA 2.30: OPCIONES ELIMINAR Y EDITAR CATEGORÍAS	81
FIGURA 2.31: BURN DOWN CHART SPRINT 5	83
FIGURA 2.32: INGRESAR PRODUCTOS.....	84
FIGURA 2.33: MODIFICAR PRODUCTOS.....	84
FIGURA 2.34: INTERFAZ ADMINISTRACIÓN DE PRODUCTOS	84
FIGURA 2.35: BURN DOWN CHART SPRINT 6.....	86
FIGURA 2.36: MENÚ PEDIDOS GENERADO EN LA SEXTA ITERACIÓN (SPRINT 6).....	87
FIGURA 2.37: OPCIÓN EDITAR PEDIDO.....	87
FIGURA 2.38: GENERACIÓN DE LA FACTURA ASOCIADA A UN PEDIDO.	88
FIGURA 2.39: REPORTE DE PRODUCTOS VISTOS ORDENADOS DESCENDENTEMENTE POR PORCENTAJE	88
FIGURA 2.40: DASHBOARD	89
FIGURA 2.41: BURN DOWN CHART SPRINT 7	91
FIGURA 2.42: INGRESAR INFORMACIÓN EMPRESA.....	91

FIGURA 3.1: PÁGINA PRINCIPAL DEL SISTEMA VISTA EN CHROME 39	124
FIGURA 3.2: PÁGINA PRINCIPAL DEL SISTEMA VISTA EN FIREFOX 33.1.....	125
FIGURA 3.3: PÁGINA PRINCIPAL DEL SISTEMA VISTA EN INTERNET EXPLORER 8	125
FIGURA 3.4: PRUEBA DE RENDIMIENTO - TIEMPO DE RESPUESTA	127
FIGURA 3.5: REPRESENTACIÓN GRÁFICA DE LAS RESPUESTAS A LA PREGUNTA 1	130
FIGURA 3.6: REPRESENTACIÓN GRÁFICA DE LAS RESPUESTAS A LA PREGUNTA 2.	131
FIGURA 3.7: REPRESENTACIÓN GRÁFICA DE LAS RESPUESTAS A LA PREGUNTA 3.	132
FIGURA 3.8: REPRESENTACIÓN GRÁFICA DE LAS RESPUESTAS A LA PREGUNTA 4.	132
FIGURA 3.9: REPRESENTACIÓN GRÁFICA DE LAS RESPUESTAS A LA PREGUNTA 5.	133

RESUMEN

El proyecto de titulación detallado a continuación consiste en el desarrollo e implementación de un sistema de comercio electrónico, del tipo B2C, para la tienda de calzado y artículos de cuero EMI.

Este documento es la evidencia del trabajo realizado y está conformado por 4 capítulos.

En el primer capítulo se proporciona el sustento teórico, necesario para entender el qué y el porqué de cada una de las prácticas llevadas a cabo durante el desarrollo y las técnicas de gestión del proyecto utilizadas. Para el desarrollo de este proyecto se ha hecho uso de herramientas libres y se lo ha gestionado con una metodología híbrida. Se escogió Scrum como metodología de gestión del proyecto, para las prácticas de programación y las técnicas de desarrollo de software se hizo uso de Extreme Programming (XP). Ambas metodologías encajan en la clasificación de metodologías ágiles.

En el segundo capítulo se especifica cada uno de los puntos que conforman el ciclo de desarrollo de software, siendo estos la Especificación de Requerimientos, el Análisis, el Diseño, la Implementación y las Pruebas. Si bien estos puntos se presentan de una manera secuencial, no está por demás aclarar que al tratarse de metodologías iterativas e incrementales, en cada iteración se atraviesa por cada una de ellas.

El tercer capítulo se centra en la futura implantación del sistema ya concebido y de las pruebas previas a dicha implantación, realizadas en un ambiente de preproducción semejante al de producción para así garantizar un despliegue limpio, libre de errores, eficiente y funcionalmente aceptado por los usuarios tanto administradores como clientes.

El capítulo cuatro recoge las conclusiones más importantes que resultaron de este proyecto, así como también las recomendaciones producto de la experiencia adquirida.

PRESENTACIÓN

En el mundo globalizado de hoy, que cada vez sumerge más sus procesos en las aguas de la tecnología, la adopción de sistemas de comercio electrónico se hace más urgente para las pequeñas y medianas empresas, a fin de que puedan promocionar sus productos y mantenerlos a la vista de quienes ya son sus clientes y quienes lo serán más adelante.

En este documento, se presenta la solución informática para la tienda de zapatos y artículos de cuero EMI, un sistema de comercio electrónico B2C, codificado con el lenguaje de programación PHP, haciendo uso de la base de datos PostgreSQL, y desarrollado en el marco del patrón MVC (Modelo-Vista-Controlador).

El sistema facilitará las tareas de administración de productos, control de stock y permitirá a sus clientes realizar compras on-line escogiendo los métodos de envío y la forma de pago.

Como más adelante se podrá evidenciar, el sistema desarrollado fue bien aceptado entre un grupo de usuarios finales, pudiendo concluir que el esfuerzo y trabajo que se han invertido en él no ha sido en vano.

1 CAPÍTULO 1. PLANTEAMIENTO DEL PROBLEMA

1.1 DEFINICIÓN DEL PROBLEMA

1.1.1 ANTECEDENTES DEL PROBLEMA

En la actualidad el Internet y la tecnología se han convertido en elementos esenciales del diario vivir, llegando a cambiar la rutina diaria de convivencia y trabajo^[69]. Las transacciones comerciales son un ejemplo de este cambio. Hoy por hoy el comercio electrónico está dejando de ser una “opción” para aquellas empresas que desean desarrollarse a un buen ritmo, ha pasado a ser un requisito pues todo aquello que ayude a ahorrar tiempo, movilización y esfuerzo son elementos básicos para la elección de productos y/o servicios por parte de los potenciales clientes.

Las empresas emprendedoras, sin duda alguna, se suman a la revolución tecnológica del comercio electrónico, lo que supone un cambio incluso en la cultura comercial que se ha acostumbrado a manejar, sin embargo, los beneficios de atreverse a este cambio son muchos y logran importantes mejoras.

El Internet brinda muchas soluciones que se adaptan a diversas realidades tanto económicas como sociales, ofreciendo sobretodo una ventana de comunicación que ayuda a difundir la información de la compañía y gracias a ello le permite a ésta aprovechar las oportunidades de llegar a nichos de mercado que tiempo atrás consideraban inalcanzables^[79].

Por este hecho, la empresa de Calzado y Artículos de Cuero EMI se ha decidido al cambio. Desea pasar del método tradicional de negocios a abrazar las oportunidades que el Internet y la tecnología le ofrecen.

^[69] TREJO DELARBRE, Raúl. Vivir en la Sociedad de la Información.

^[79] América economía. Estudio de comercio electrónico en América Latina.

1.1.2 COMERCIO ELECTRÓNICO

Al comercio electrónico se lo puede considerar, en términos sencillos, como la compra-venta de productos y servicios a través de sistemas electrónicos, principalmente: Internet. ^[104]

Varios son los factores que hoy están impulsando el crecimiento de negocios digitales, entre ellos la banda ancha, la banda móvil, incluso el crecimiento de las televisiones digitales inteligentes que se cuentan como el tercer dispositivo de acceso a internet (después del computador y los tablets/smartphones). Sin embargo, el verdadero despegue del comercio electrónico y la inclusión del mismo en la economía de todo el mundo se dió desde la puesta en marcha de Internet, la cual tuvo una aparición progresiva pero significativa, ya que de ser un proyecto militar ha pasado a ser un instrumento de suma importancia en el día a día.

La inclusión de la electrónica en el comercio, en especial en el comercio internacional, ha provocado que las industrias den un giro en su forma de actuar para aprovechar las nuevas oportunidades, incluyendo los nuevos canales de comercialización de productos existentes y desarrollando un nuevo tipo de información. Es decir, la implantación del comercio electrónico implica la adopción de una nueva filosofía como empresa, determinada por la nueva realidad en el intercambio de productos pues ya las barreras geográficas, la distancia y el tiempo no suponen problema alguno, pero deben ser contempladas en el diseño de la solución electrónica a adoptar. ^[75]

Es por esto que esta utilización de las nuevas tecnologías y su aplicación al mundo de los negocios conlleva a un replanteamiento de los roles de los agentes económicos, pues en ésta el cliente no sólo es el centro de todas las acciones, sino que abandona su papel de receptor de información y se convierte en un consumidor activo y con una mayor oferta.

^[104] Wikipedia. Comercio electrónico.

^[75] Asociación Española de la Economía Digital (adigital). Libro blanco del comercio electrónico.

1.1.2.1 Definiciones De Comercio Electrónico

Entre la amplia cantidad de definiciones existentes para describir qué es el Comercio electrónico, se recopila a continuación las más representativas:

Según lo expresa Mateu de Ros, el comercio electrónico constituye un fenómeno jurídico y se concibe como la oferta y la contratación electrónica de productos y servicios a través de dos o más ordenadores o terminales informáticos conectados a través de una línea de comunicación dentro del entorno de red abierta que constituye Internet. Representa un fenómeno en plena expansión con votos de crecimiento extraordinario en número de conexiones, clientes y operaciones. ^[45]

Por su parte, Renato Javier Jijena Leiva, dice que el comercio electrónico es el intercambio telemático de información entre personas que da lugar a una relación comercial, consistente en la entrega en línea de bienes intangibles o de un pedido electrónico de bienes tangibles, que pueden ser multimedia, es decir, consistir en imágenes, textos y sonidos. ^[29]

Para Mario de la Gaza: “El comercio electrónico viene a ser un envoltorio conjunto de herramientas de tecnologías de información. Así como estrategias de negocios destinadas a favorecer la realización de prácticas comerciales de forma electrónica. Cabe señalar que también el término comercio electrónico se usa para designar las operaciones que personas, empresas organizaciones y gobiernos efectúan en línea, por medio de tiendas virtuales o portales electrónicos”. ^[45]

1.1.2.2 Clases De Comercio Electrónico

La proliferación de empresas y consumidores ha diversificado los tipos de comercio electrónico, los mismos que han surgido como una necesidad de adaptarse al mercado.

^[45] NIETO MELGAREJO , Patricia. Nociones generales sobre el comercio electrónico.

^[29] JIJENA LEIVA, Renato. Comercio, derecho, firma y documentos digitales o electrónicos.

Una de las formas de clasificar al comercio electrónico, y la aquí usada, es según la participación de los sujetos o agentes económicos que intervienen. ^[47]

1.1.2.2.1 Comercio Electrónico B2B (Comercio Entre Empresas)

Solución de comercio electrónico entre empresas (fabricantes, distribuidoras y comercios).

El comercio electrónico B2B (Business to Business) es el negocio orientado entre las diversas empresas que operan a través de Internet^[3].

Dentro del comercio electrónico B2B se pueden distinguir tres modalidades:

- El mercado controlado por el vendedor en busca de compradores.
- El mercado controlado por el comprador que busca proveedores.
- El mercado controlado por intermediarios que persiguen el acuerdo entre vendedores y compradores.

El comercio electrónico B2B ha supuesto un gran avance tecnológico, pero se requieren una serie de características para obtener el rendimiento óptimo:

- Experiencia en el mercado específico.
- La oferta debe ser un valor añadido.
- Evitar fallos de producción, logística y distribución.

Las ventajas y características han convertido al comercio B2B en una opción que cada vez tienen más adeptos.

1.1.2.2.2 Comercio Electrónico B2C

El comercio electrónico B2C (Business to Consumer) viene a ser la transacción a través de medios electrónicos como Internet, que se realiza entre la empresa y el consumidor final^[85].

^[47] OLTRA, Juan V. Clasificaciones del comercio electrónico.

^[3] ANDAM, Zorayda Ruth. E-Commerce and e-Business.

^[86] Luxor Technologies. B2C Business to Consumer.

Las ventajas más destacables del comercio electrónico B2C son:

- Compra más cómoda y más rápida.
- Ofertas y precios siempre actualizados.
- Centros de atención al cliente integrados en la web.

Existen diferentes tipos de comercio electrónico B2C:

- **Intermediarios on-line:** se trata de compañías que facilitan las transacciones entre compradores y vendedores a cambio de una contraprestación económica a modo de porcentaje.
- **Modelos basados en la publicidad:** publicidad basada en el sistema, donde las empresas tienen sitios web de un inventario, que venden a las partes interesadas.
- **Modelos basados en la comunidad:** las empresas facultan a los usuarios para interactuar, en todo el mundo, en áreas similares de interés.
- **Modelos basados en tarifas:** es un sistema de pago basado en el sistema. En este caso la empresa cobra una tarifa de suscripción para ver su contenido.

1.1.2.2.3 Comercio Electrónico B2A

El comercio electrónico B2A (Business to Administration) es un servicio que ofrece la administración a las empresas –y también a los ciudadanos– para que se puedan realizar los trámites administrativos a través de Internet^[41].

Las ventajas para las empresas son evidentes:

- Ahorro considerable de tiempo y esfuerzo.
- La posibilidad de descargarse formularios y modelos de los procedimientos administrativos.
- Disponibilidad las 24 horas del día.
- Información siempre actualizada.

^[41] MARTÍNEZ Esther. E-Business para las PYMES. 2003

1.1.2.2.4 Comercio Electrónico B2E

El comercio electrónico B2E (Business to Employee) es otra aplicación que, en este caso, relaciona a las empresas con sus empleados. A través de la intranet el empleado puede ejercer parte de sus funciones de los procesos de negocio de la empresa.

El comercio electrónico B2E ofrece ventajas significativas:

- Menores costes y burocracia.
- Formación en línea.
- Mayor calidad en la información interna.
- Equipos de colaboración en el entorno web.
- Integración más ágil del profesional en la empresa.
- Soporte para la gestión.
- Comercio electrónico interno.
- Fidelización del empleado.

1.1.2.2.5 Comercio Electrónico C2C

El comercio electrónico C2C (Consumer to Consumer) es el tipo de comercio que se lleva a cabo entre consumidores, bien sea mediante el intercambio de correos electrónicos o a través de tecnologías P2P (peer to peer).

Una de las estrategias más comunes del comercio C2C para Internet viene definida por aquel tipo de negocio cuyo objetivo es facilitar la comercialización de productos y/o servicios entre particulares. Ejemplos muy conocidos son e-bay o mercadolibre.com. ^[99]

1.1.2.2.6 Comercio Electrónico C2G

El comercio electrónico C2G (Citizen to Government) relaciona a los consumidores con el Gobierno, facilitando el intercambio telemático de transacciones entre los ciudadanos y las administraciones públicas.

^[99] Sonja Rosúa Clyne. Cómo funciona el modelo de comercio electrónico C2C (Consumer to Consumer).

Algunos de los servicios más habituales son:

- Información.
- Participación del ciudadano.
- Suscripción para la notificación telemática.
- Pago de tasas e impuestos.
- Sugerencias y reclamaciones.
- Entrada y/o salida a través de registro.
- Diversos servicios, como empleo, sanidad o educación.

1.1.2.2.7 Comercio Electrónico B2G

El comercio electrónico B2G (Business to Government) busca una mejor optimización de los procesos de negociación entre empresas y el gobierno. Su aplicación se destina a los sitios o portales especializados en la administración pública. En ellos las instituciones oficiales tienen la posibilidad de contactar con sus proveedores, pudiendo estos agrupar ofertas o servicios^[55].

1.1.3 JUSTIFICACIÓN DEL MODELO DE COMERCIO ELECTRÓNICO APLICADO AL PROYECTO

Una vez descritos los diferentes tipos de comercio electrónico, se concluye que el B2C es el que encaja con la naturaleza del proyecto a desarrollar, pues el sistema a crearse, denominado ESIWECOM, permitirá realizar la transacción compra-venta entre el consumidor final y la empresa de calzado y artículos de cuero EMI.

1.1.4 FACTIBILIDAD DEL PROYECTO

Un proyecto es factible cuando la ejecución de varias actividades íntimamente ligadas entre sí da como resultado el logro de objetivos previamente definidos en atención a las necesidades planteadas, es decir, para las necesidades de la tienda EMI. La finalidad del proyecto factible radica en el diseño de una propuesta de acción dirigida a resolver un problema o necesidad previamente detectada en el medio.

^[55] TASSABEHJI, Rana. Understanding E-commerce for Business.

A fin de concluir si el proyecto es o no factible, se detallan a continuación los objetivos que se consideran alcanzables con la consecución del sistema.

- Reducción de errores y mayor precisión en los procesos.
- Actualización y mejoramiento de los servicios a clientes o usuarios.
- Aceleración en la recopilación de datos.
- Reducción en el tiempo de procesamiento y ejecución de tareas.
- Automatización óptima de procedimientos manuales.

En este caso en particular, se analizará la factibilidad del proyecto en la siguiente clasificación:

1.1.4.1 Factibilidad Técnica

En este apartado se determinará si se dispone de los conocimientos, habilidades y herramientas para llevarlo a cabo el proyecto. Se lo hará analizando la capacidad de mejora del sistema actual, y la disponibilidad de la tecnología necesaria para este fin^[52].

1) Mejora de la situación actual.

Es totalmente factible lograr una mejora sustancial del sistema con respecto a su situación actual, pues al momento todo el proceso es manual, trayendo como consecuencia la introducción de errores en su mayoría involuntarios, causados por una escasa o nula ayuda de las herramientas tecnológicas.

2) Disponibilidad de tecnología que satisfaga las necesidades

Para desarrollar el sistema web de comercio electrónico, se necesitará básicamente el uso de una arquitectura web, la misma que incluye un servidor web, aplicaciones web y servicios web. Todos éstos son alcanzables, ninguno supone dificultad alguna en la consecución del sistema.

[52] PINEDA MORENO , Milena. Estudio de factibilidad sobre el desarrollo de software a la medida de marcación predictiva. Colombia. 2007.

1.1.4.2 Factibilidad Económica

Corresponde al estudio de los recursos tiempo y costo.

1) **Tiempo del analista/equipo de desarrollo.**

Este punto queda saldado en virtud de un análisis de requerimientos previo que sea realista y que permita calendarizar las tareas a fin de conseguir el objetivo trazado en un tiempo prudencial^[8].

2) **Costo del tiempo del personal.**

En la situación actual, el costo del sistema no supone desventaja para la empresa que se verá beneficiada de este trabajo que se desarrollará a manera de tesis de grado.

3) **Costo del desarrollo / adquisición**

Las herramientas que serán necesarias para el desarrollo del sistema no tienen costo, ya que la estrategia de implementación se basa en el uso de software libre. Las razones para esta elección son explicadas en el siguiente punto de este documento.

Por otro lado, el servicio que si deberá ser cubierto en este desarrollo es el de alojamiento. A este respecto la decisión por la que se ha optado se enfoca en el concepto de “Plataforma como servicio”, que en base al hosting VPS (Virtual Private Server) hace posible acceder a la infraestructura necesaria a bajos precios. En el capítulo 3 se explicará con más detalle en qué consiste la oferta de VPS.

En la red, se puede encontrar servicios de VPS disponibles a solo una transacción electrónica, sea por cuenta bancaria o PayPal. En la Tabla 1.1 se recogen algunos de las ofertas que hoy por hoy existen con respecto a este servicio.

^[8] BOEHM, Barry W.. Software Engineering Economics. IEEE. 1984.

HOSTING VPS	PRECIO	DISCO	RAM	TRANSFERENCIAS
Cloud VPS	\$16,95 /mes	250 GB	2 GB	Ilimitadas
Servicom	\$250/anual	Ilimitado	4 GB	200 GB
OpenVZ Linux VPS	\$45/mes	1000 GB	2 GB	Ilimitadas
Hosteandote	\$124,95/mes	30 GB	512 MB	250 GB

Tabla 1.1

: COMPARACIÓN ENTRE DIFERENTES SERVICIOS DE HOSTING VPS

Como lo muestra la Tabla 1.1, los precios son variados y al alcance de cualquier bolsillo, por lo que se puede aseverar que este proyecto es económicamente factible.

1.1.5 BENEFICIOS DEL PROYECTO

Los beneficios que se derivan de adoptar una solución tecnológica para la promoción y venta de los artículos que ofrece a su mercado la tienda EMI, se detallan a continuación^{[54][55]}:

- Incrementar en número los clientes de la tienda para lo cual será de ayuda la publicidad que se le pueda dar al sitio, en un principio, a través de las redes sociales.
- Popularizar los productos que la tienda ofrece mediante estrategias visuales propias del desarrollo web.
- Generar un mayor número de ventas por cliente a través de las promociones que podrán visualizarse en el sistema.
- Mejorar los canales de comunicación con el público al que va dirigida la oferta al aceptar sugerencias de los usuarios del sistema, que serán los clientes de la tienda.
- El beneficio que se deriva como consecuencia de los anteriores será el incremento de las ganancias que obtenga la tienda EMI por la venta de sus productos.

^[54] Puromarketing.com. Internet y e-commerce: las ventajas de las tiendas online para su negocio. 2012.

^[55] TASSABEHJI, Rana. Understanding E-commerce for Business.

1.2 SELECCIÓN DE LA METODOLOGÍA DE DESARROLLO

La ingeniería de software tiene varias metodologías o paradigmas de desarrollo, que sirven de apoyo para la realización de proyectos de software, y dependiendo de las características del proyecto, se deberá inclinar en la utilización de una u otra opción; que a posterior, se traducirá en el éxito o fracaso del proyecto.

En la actualidad, el aparecimiento de nuevas propuestas metodológicas, conjuntamente con aquellas de larga trayectoria; ambas con ventajas y desventajas, se encuentran en constante debate, lo que provoca incertidumbre, llevando a cuestionarse: ¿Qué metodología de desarrollo de software se debe utilizar en el proyecto? ^[20].

A partir de esta premisa y bajo la consideración de los autores de este proyecto, se realizará una corta comparación y análisis según los criterios de clasificación descritos en el presente trabajo; para una ulterior elección e implementación de una metodología que se ajuste a las necesidades del proyecto.

1.2.1 METODOLOGÍAS TRADICIONALES

De acuerdo a la filosofía de desarrollo de las metodologías, aquellas que son mucho más estrictas en la planificación y control del proyecto, en la especificación de los requerimientos, reciben el nombre de “Metodologías Tradicionales”.

Estas metodologías aplican un enfoque disciplinado de trabajo en todo el proceso de desarrollo de software, en las etapas de análisis, planificación, diseño e implementación del proyecto, con el objetivo de obtener un producto de software eficiente^[63].

Estas metodologías son muy bien documentadas, con un alto nivel de detalle, y por tanto, llegan a ser bastante complejas de aplicar; con lo que la gestión misma de estas metodologías, conlleva a un trabajo aún mayor que el propio desarrollo del producto de software. Además, respecto a los roles, actividades y demás

^[20] BRITO ACUÑA, Karenni. Selección de Metodologías de Desarrollo para Aplicaciones Web.

^[63] SOMMERVILLE , Ian. Ingeniería del Software.

componentes que participan en el control del proceso de desarrollo, son definidos de manera inflexible, lo que no beneficia a proyectos que requiriesen algún cambio en determinada fase del desarrollo^[38].

Las fases de análisis y diseño son las principales etapas en los enfoques tradicionales. Todos los enfoques tradicionales son métodos sistemáticos de desarrollo de software, en donde, una vez que el producto de software está bien diseñado, el proyecto puede continuar con la fase de implementación, lo que supone será predecible.

En ese escenario, se pueden ver dos actividades principales:

- El diseño, que es una actividad con un alto grado de creatividad, difícil de predecir y que sigue siendo la parte principal del proceso de desarrollo de software.
- La implementación, que es fácil de predecir, pues se sigue lo ya diseñado, y que viene a ser una parte más pequeña del proceso de desarrollo de software.

Por ello, las metodologías tradicionales de desarrollo de software son útiles en algunos proyectos y carentes de éxito en otro tipo de proyectos. Debido a que su principal característica es el detallado análisis y diseño, son útiles cuando el proyecto se dice que es muy grande y el nivel de riesgo es alto; en algunos proyectos en que se estima que durará mucho tiempo se desarrollan a menudo con alguna metodología tradicional. ^[71] ^[28] Los equipos de trabajo son integrados por una cantidad extensa de roles, y cada uno cumple estrictamente con su función.

Los enfoques tradicionales, en general, se pueden describir por sus características principales:

- Son notables cuando el proyecto es grande y la conformación de los equipos de desarrollo también es grande.

^[38] LEÓN SERRANO, Gonzalo. Ingeniería de Sistemas de Software.

^[71] WYSOCKI, Robert, Effective Project Management - Traditional, Agile, Extreme.

^[28] JALOTE, Pankaj, An Integrated Approach to Software Engineering.

- La planificación del proyecto es muy detallada.
- Posee una amplia documentación en donde se describen todas las características del sistema.
- Se utilizan en proyectos en donde los requerimientos principales son definidos inicialmente y no cambian con frecuencia (lo que contempla una rigidez a posibles cambios en el transcurso del desarrollo).
- Las fases de análisis, planificación, diseño, implementación y despliegue son consecutivas.

Entre las Metodologías Tradicionales más representativas, podemos citar a dos de ellas:

- Rational Unified Process (RUP).
- Microsoft Solution Framework (MSF).

Éstas se caracterizan por concentrar sus esfuerzos en mantener una documentación íntegra de todo el proyecto, y en el cumplimiento con la planificación definida en la fase inicial del proceso de desarrollo, lo que da a estas metodologías el calificativo de predictivas^[21].

1.2.2 METODOLOGÍAS ÁGILES

Las Metodologías Ágiles, se enfocan en la adaptabilidad a los cambios, de esta manera, las necesidades del cliente que pueden sufrir cambios imprevistos en proyectos en los que los requisitos son desconocidos o variables, obtienen respuestas rápidas lo que favorece a la adaptabilidad de los proyectos, y el desarrollo de software resulta ser mucho más rápido y eficiente^[32].

Una Metodología Ágil, facilita la generación pronta de entregables funcionales, previos a la entrega del producto de software final, lo que para el cliente por la premura del tiempo, puede resultar muy beneficioso. El desarrollo con estas

^[21] Roberth Figueroa, Camilo Solís, Armando Cabrera. Metodologías Tradicionales Vs. Metodologías Ágiles.

^[32] Nayan Jyoti Kar. Adopting Agile Methodologies of Software Development.

metodologías es iterativo e incremental; con cada entregable, se obtendrá un producto de software más maduro.

Se han desarrollado una lista extensa de metodologías que se acogen a esa filosofía, dentro de las cuales podemos listar algunas de ellas:

- Adaptive Software Development (ASD).
- Agile Unified Process (AUP).
- Crystal Clear.
- Feature Driven Development (FDD).
- Open Unified Process (OpenUP).
- Extreme Programming (XP).
- Scrum.

1.2.3 COMPARACIÓN ENTRE METODOLOGÍAS DE DESARROLLO

A manera de resumen, en la Tabla 1.2 se enmarca el enfoque tanto de las Metodologías Ágiles como las Tradicionales frente a un determinado conjunto de características importantes en el ámbito del desarrollo de software.

TABLA COMPARATIVA ENTRE METODOLOGÍAS DE DESARROLLO DE SOFTWARE		
CARACTERÍSTICA	TRADICIONALES^[58]	ÁGILES^[33]
CAMBIOS	Ofrece cierta resistencia a los cambios	Preparado para cambios durante el proyecto
GRUPO DE TRABAJO	Más roles. Responsabilidades estrictas.	Pocos roles. Conocimiento colectivo.
PLANIFICACIÓN	Predictiva	Adaptativa
REQUERIMIENTOS	Estrictos, desde la fase inicial del proyecto.	Adaptables, según se vaya realizando mejoras, pueden existir posibles variaciones.
DOCUMENTACIÓN	Documentación exhaustiva y detallada.	Documentación en caso de requerirse.
CLIENTE	Es un actor externo al grupo de trabajo.	Forma parte del grupo de trabajo del proyecto.
RIESGO	Por su enfoque predictivo, si no se realiza un análisis correcto al inicio del proyecto, se pueden presentar problemas en el transcurso del mismo.	Por su enfoque adaptativo, en cada iteración se tiende a reducir los riesgos, debido a la constante retroalimentación.
CONTRATO	Se cumple estrictamente con lo planificado inicialmente.	Existe flexibilidad, y se llega a un acuerdo con el cliente para establecer parámetros no fijos.

Tabla 1.2: COMPARACIÓN ENTRE GRUPOS DE METODOLOGÍAS DE DESARROLLO

^[58] SCHACH Stephen. Object-Oriented and Classical Software Engineering.

^[33] COCKBURN Alistar. Agile Software Development.

1.2.4 DESCRIPCIÓN DE LA METODOLOGÍA ESCOGIDA (HIBRIDA: SCRUM Y XP)

Se estima que el proceso adecuado para este tipo de desarrollo de software sería un proceso ágil; que se adapte al desarrollo de diversos tipos de aplicaciones para la web (comercio electrónico, redes sociales, blogs, CMS, etc.), dando lugar a la agregación de nuevos requerimientos y sin demandar de una excesiva documentación.

Se decide utilizar dos de las metodologías más implementadas en desarrollos ágiles, y éstas son:

- Scrum.
- Extreme Programming (XP).

Para la implementación de Scrum con XP, se empleará a Scrum como metodología base, es decir, se respetará su implementación por completo sin realizar ninguna modificación a esta metodología. Sus roles, artefactos y ceremonias o actividades, se los seguirá tal cual se describen en Scrum, con lo que controlaremos la gestión del equipo de trabajo y del proyecto en general. Pero ya en cada Sprint, el desarrollo del sistema será complementado con algunas de las mejores prácticas y técnicas de Extreme Programming (XP), que permitan el aseguramiento de la calidad del producto de software que se desarrollará.

Scrum es conocida como una metodología no exclusivamente para proyectos de desarrollo de software, sino también de otra índole; es una metodología que permite gestionar de manera óptima a los grupos de trabajo, independientemente de la naturaleza del proyecto, en donde el feedback o retroalimentación iterativo constituye el elemento fundamental.

Extreme Programming (XP) por su parte, es una Metodología de Desarrollo de Software, enfocada principalmente en el desarrollo de software como tal, mediante el uso de las mejores prácticas e implementando constantemente

pruebas que aseguren la correcta ejecución de las funcionalidades que se van desarrollando.

Ambas metodologías procuran poner más énfasis en la adaptabilidad a los cambios, y por los escenarios en los que tienen mayor protagonismo estas metodologías se complementan entre sí.

Por consiguiente, se describirá los roles, artefactos y actividades de Scrum, tal cual serán implementados en la metodología híbrida que se utilizará.

Es necesario mencionar que además de la utilización tanto de XP como de Scrum a lo largo de este proyecto, se hará uso de diagramas de navegabilidad para la documentación del diseño.

1.2.4.1 Roles De Scrum

Como ya se especificó, para el presente proyecto se implementará Scrum en su totalidad, por lo que se utilizarán los roles que presenta Scrum, los cuales son^[15]:

Product Owner: persona responsable de transmitir los intereses del cliente respecto al producto que requiere para su organización^[98]. Puede ser un usuario final, un consumidor del producto, un usuario clave u otro, perteneciente o no a la organización pero que está plenamente impregnado de las características y funcionalidades con las que deberá cumplir el producto de software a desarrollarse;

Scrum Master: es quien lidera al equipo de trabajo dentro del proyecto, y entre sus responsabilidades está el de facilitar que las reuniones de Scrum se realicen de manera planificada^[50];

Team: es el grupo de personas que desarrollan el producto de software del proyecto

[15] DEEMER, Pete, BENEFIELD Gabrielle, LARMAN , Craig, Bas Vodde. The Scrum Primer.

[98] ScrumSense. What every Product Owner should know.

[50] PHAN, Andrew, PHAM, Phuong-Van. Scrum in Action: Agile Software Project Management and Development.

1.2.4.2 Artefactos De Scrum

Scrum propone tres artefactos o herramientas para organizar los proyectos, estos ayudan a mantener una planificación y revisión dentro de cada Sprint.

Los artefactos mencionados son los siguientes^[7]:

- Product Backlog.
- Sprint Backlog.
- Burndown Chart.

La definición de estos artefactos se hará en el Capítulo 2 al tratar el Diseño del sistema.

1.2.4.3 Actividades De Scrum

Scrum es una metodología en la que se aplican una serie de buenas prácticas para trabajar de manera colaborativa, y en equipo, con el fin de obtener como resultado, un proyecto de calidad. ^[96] De manera gráfica se puede observar el proceso de Scrum en la Figura 1.1.

1.2.4.3.1 Sprint Planning Meeting (Planificación De La Iteración)

Al comienzo del ciclo del Sprint o iteración (cada 2 a 4 semanas), se realiza una reunión de planificación del Sprint.

En esta reunión se lleva a cabo las siguientes actividades:

- Se selecciona el trabajo que se va a realizar.
- Se prepara el Sprint Backlog, en el que se detalla el tiempo que se necesita para hacer ese trabajo, con todo el equipo.
- Se identifica y comunica la cantidad de trabajo que probablemente se realice durante el sprint actual.

^[7] BELKINA Maria, SCHILLER, Jennifer, MASUNOV, Maxim, FILIPPOV, Vyacheslav. Scrum.

^[96] KUMAR, Sathees. What is Agile Scrum?.

- Para esta reunión, se pone un límite de 8 horas, que se dividen en dos partes de 4 horas cada una^{[49][100]}:

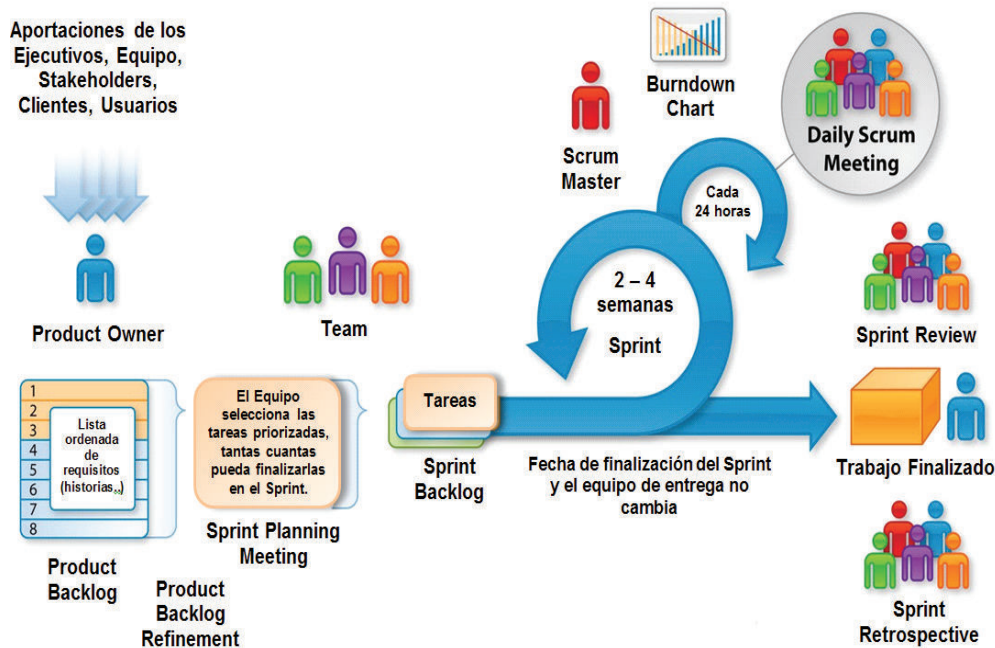


Figura 1.1: PROCESO DE SCRUM

1.2.4.3.2 Sprint (Iteración)

Un Sprint es la unidad básica de desarrollo de Scrum con una duración específica, y se realiza en periodos cortos y fijos, que comprende a iteraciones que van de 2 a 4 semanas en las cuales se alcanza un incremento funcional del producto que se está desarrollando. ^[84]

[49] PALACIO, Juan. Scrum Manager - Manual - Gestión de Proyectos.

[100] HAMILTON-WHITAKER, Tara. Scrum Sprint Planning Meetings – Who, What, When, Where, Why.

[84] SZALVAY, Laszlo. Scrum Sprint.

1.2.4.3.3 Scrum Daily Meeting (Reunión Diaria De Scrum)

Reunión diaria, de máximo 15 minutos en la que cada miembro del Equipo responde tres preguntas^[73]:

- ¿Qué ha hecho desde la última reunión?.
- ¿Qué planea hacer desde este momento?.
- ¿Qué impedimentos tuvo o va a tener para cumplir con los compromisos en esta iteración y en el proyecto?.

1.2.4.3.4 Sprint Review (Demostración De Requisitos Completados)

El Sprint Review brinda un punto de inspección o revisión para verificar el progreso del proyecto al final de cada Sprint. En esta reunión que tiene como duración un máximo de 4 horas, el Equipo presenta al Product Owner las metas del Sprint, con su respectivo Product Backlog comprometido y las funcionalidades que se completaron, más no las que no se hayan culminado. ^[48]

1.2.4.3.5 Sprint Retrospective (Reunión De Retrospectiva)

Posterior a la reunión de Sprint Review, se realiza la reunión de Sprint Retrospective, en la que se discute el Sprint recientemente finalizado, y se determina los cambios que se realizarán en el siguiente Sprint según lo que ya se demostró y discutió con el Product Owner anteriormente. Como resultado se obtiene un feedback (retroalimentación), que puede aumentar la productividad del equipo, y consecuentemente la calidad del producto. ^[95]

1.2.4.3.6 Product Backlog Refinement O Product Backlog Grooming (Replanificación Del Proyecto)

El objetivo de esta reunión es de ayudar al Product Owner a tener lista la parte superior del Product Backlog para la próxima reunión de Sprint Planning. El Product Owner no puede hacer esto por sí sólo sino que todo el Equipo debe

^[73] MAGNO, Alexandre. The Daily Meeting Trap.

^[48] PALACIO, Juan, RUATA, Claudia. Scrum Manager - Gestión de Proyectos.

^[95] VAN SOLINGEN, Rini. How to improve your Sprint Retrospective meeting?.

contribuir, sin embargo, la decisión final de los requisitos, en especial la priorización, la realiza el Product Owner^[74].

1.2.4.4 Prácticas XP En Scrum

Ya establecido el proceso de Scrum, en cada Sprint que se haya planificado, al momento de desarrollar las funcionalidades acordadas, se hará uso de prácticas de XP^[5], de las cuales se utilizará 7 de ellas, las que serán descritas en los siguientes puntos.

En la Figura 1.2, se puede observar las prácticas XP que se incluyen en la metodología híbrida que se utilizará basada en Scrum con XP.

1.2.4.4.1 Diseño Simple

Lo que se busca no es un diseño predictivo del sistema a desarrollar, sino lo que se defiende es realizar un diseño en cada momento para las necesidades que se presenten, manteniendo lo más simple que sea posible. De esta forma, siempre será mucho más rápido y barato implementar un diseño simple en lugar de uno complejo.^[76]

Como recomendación, no se debe adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se está trabajando.

^[74] GOTHE , Anand. Importance of Product Backlog Refinement in Scrum.

^[5] BAHIT, Eugenia. Scrum y Extreme Programming para programadores.

^[76] WELLS, Don. Simplicity is the Key.

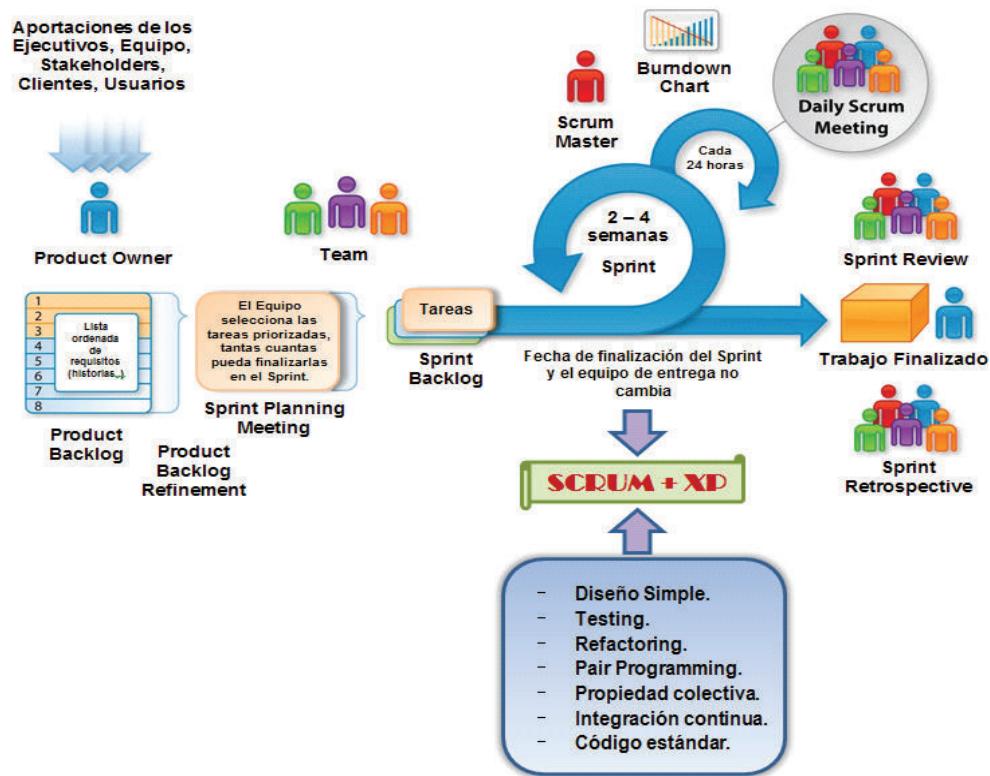


Figura 1.2: PROCESO SCRUM CON XP

1.2.4.4.2 Testing / TDD

Es el pilar fundamental sobre el que se sustenta XP, es decir, si no existen tests, no se estará aplicando XP^[6]. Para esto se tendrá que emplear frameworks que permitan automatizar el testing, como PHPUnit, etc. Para eso, estas pruebas o tests, deberán ser diseñadas antes de incluso haber programado algo. El cómo escribir este tipo de pruebas se puede encontrar más adelante en el apartado correspondiente a Pruebas Unitarias.^[31]

Es necesario determinar las clases más importantes del sistema para darles prioridad en el desarrollo de las pruebas unitarias, esto debido a que codificarlas es una tarea que demanda una inversión de tiempo igual o superior a la que toma codificar las clases del sistema propiamente dichas.

^[6] BECK, Kent. Extreme Programming Explained.

^[31] JURADO, Carlos Blé. Diseño Ágil con TDD.

1.2.4.4.3 Refactoring

Responde al principio de simplicidad, y consiste en que el código escrito debe dejárselo en la forma más simple posible, que no pierda ni incremente su funcionalidad, y que se ejecute correctamente en todos los tests. Esto a futuro ayudará a desarrolladores que no hayan participado del desarrollo inicial y que tengan que realizar cambios o añadir funcionalidades, a que no tengan que tomar mucho tiempo para entender el código escrito por otros programadores.^[70]

1.2.4.4.4 Pair Programming (Programación Por Parejas)

Se caracteriza porque el código será desarrollado en parejas, es decir, compartiendo un solo equipo para el desarrollo, en donde, mientras el uno codifica pensando en la mejor forma de implementarlo, el compañero lo hará de una manera más estratégica, apoyando a su compañero en caso de que se presentara algún inconveniente, por lo que ambos son los autores del código. Los roles por supuesto son intercambiables, con lo que el que estaba observando puede pasar a codificar^[56].

1.2.4.4.5 Propiedad Colectiva

Cualquiera de los miembros del equipo puede modificar cualquier parte del código, en cualquier momento, claro está, siempre que haya antes escrito las pruebas correspondientes. Para esto, se hace necesario el uso de estándares de codificación, y la utilización de tests que permitan asegurar que va a seguir funcionando bien las aplicaciones, aún después de una modificación^[61].

1.2.4.4.6 Integración Continua

En cada día de programación, se estará integrando el sistema completo, en un entorno de “integración”. En ese entorno los programadores cada vez que tengan una parte del sistema, y que haya sido probada unitariamente, añadirán esa nueva funcionalidad al sistema completo. En el caso de que el sistema siga

^[70] WAKE , William. Extreme Programming Explored.

^[56] HERRANZ, Raúl. Scrum Manager - En busca de la excelencia del código.

^[61] SHORE James, WARDEN, Shane. The Art of Agile Development.

funcionando correctamente luego de realizar las respectivas pruebas unitarias, los programadores darán por finalizada esa tarea. Caso contrario, tendrán que responsabilizarse en dejar el sistema de nuevo con los tests funcionando por completo. Si no fuera posible identificar el error luego de cierto tiempo, se desechará el código a la basura y se empezará de nuevo.

1.2.4.4.7 Código Estándar

Se debe procurar mantener un estándar en la escritura del código fuente, pues esto permitirá tener un código más legible y limpio, lo que facilitará el trabajo a otros programadores que pudieran participar del desarrollo. ^[6] ^[40]

Para el caso de PHP que será el lenguaje a utilizar, el Equipo es quien deberá ponerse de acuerdo en definir y documentar sus propias reglas, aunque se puede referenciar algunas buenas prácticas de estándares conocidos como el Zend.

Estas siete prácticas de Extreme Programming (XP) descritas anteriormente, serán las que se implementarán en cada Sprint de Scrum, para el desarrollo de nuestro proyecto.

1.3 JUSTIFICACIÓN DE LAS HERRAMIENTAS DE DESARROLLO

1.3.1 SOFTWARE LIBRE

La elección del software libre se da tanto por la gratuidad como por la libertad, siendo éste último aspecto el más importante.

Si bien el no pago de licencias es un punto decisivo, el software libre presta varios beneficios que hacen de él la decisión correcta para nuestro sistema:

- No es propenso al ataque de virus - troyanos.
- Permite la fácil instalación de paquetes necesarios para el desarrollo de sistemas. Esto debido a que la mayoría de distribuciones Linux incorporan algunos de los sistemas que unifican el método de instalación de programas,

^[6] BECK, Kent. Extreme Programming Explained.

^[40] MARTIN, Robert C.. Clean Code – A Handbook of Agile Software.

librerías, etc. por parte de los usuarios, permitiendo el acceso a las miles de aplicaciones existentes de forma segura y gratuita evitando tener que acceder a páginas web de dudosa ética que pueden resultar perjudiciales para los sistemas.

- El software libre, a través de sus preceptos, evoca realmente la libertad, no la quita.
- La libertad de ejecutar el programa para cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para ayudar a su prójimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.
- El software libre ofrece independencia tecnológica pues el acceso al código fuente permite el desarrollo de nuevos productos sin la necesidad de desarrollar todo el proceso partiendo de cero.
- Otro aspecto importante es el fomento de la libre competencia al basarse en servicios y no licencias. Uno de los modelos de negocio que genera el software libre es la contratación de servicios de atención al cliente. Este sistema permitirá que las compañías que den el servicio compitan en igualdad de condiciones al no poseer la propiedad del producto al cual dan el servicio.
- El software libre estandariza los formatos lo que permite una interoperabilidad más alta entre sistemas, evitando incompatibilidades.

1.3.2 GESTORES DE BASES DE DATOS

La base de datos escogida es PostgreSQL, pues se considera que su uso supone algunas ventajas:

- Es un motor potente de base de datos cuya mayor fortaleza es la escalabilidad, ya que permite un almacenamiento de grandes cantidades de datos. Cuando la base ha crecido lo suficiente, en comparación con otros

motores, PostgreSQL requiere un menor grado de mantenimiento, mismo que puede llegar a ser complejo con otras bases libres o propietarias.

- Ofrece la facilidad de efectuar una programación robusta a nivel de base de datos, pues cuenta con el lenguaje PLPGSQL que permite la creación de procedimientos, funciones, disparadores con un nivel superior en complejidad (al compararlo con otras bases libres de menores prestaciones) lo que a su vez se traduce como mayor seguridad a nivel de datos.
- Al usar el cliente PGAdmin, postgres presenta una interfaz gráfica bastante intuitiva y amigable, lo que siempre es necesario para mejorar el ambiente de programación.
- Si bien la velocidad de procesamiento de datos, al hablar de pocas cantidades de registros, es superior en motores de Base de datos como MySQL o SQLServer, las diferencias son mínimas, y totalmente contrapuestas a la velocidad que supone el procesamiento de grandes cantidades de registros, pues en este escenario PostgreSQL resulta vencedor.

1.3.3 LENGUAJE PHP

En la actualidad, se puede percatar que a nivel mundial, la mayor parte de sitios web a los que visitamos, están desarrollados en PHP, con lo cual, esto da una idea de que este lenguaje de programación web, brindan varios beneficios, entre los que se puede destacar:

PHP es soportado por una comunidad muy extensa de desarrolladores a nivel mundial, lo que conlleva a encontrar de manera mucho más fácil documentación, código fuente, foros, y demás ayudas que permitirán solventar posibles problemas que se presenten en las aplicaciones.

La curva de aprendizaje es mucho más corta en relación a otros lenguajes, sólo hay que familiarizarse con su sintaxis, si es que ya se tiene previos conocimientos de programación.

Respecto a rendimiento, PHP está utilizando un servidor muy básico, responde de manera óptima a las miles o millones de peticiones que pueden darse en un día.

Claro está que eso también dependerá de que tan buena sea la aplicación que se ha programado.

PHP al ser software libre, no se necesita pagar licencia para su utilización, y se lo puede descargar libremente desde su página oficial. Adicionalmente, ya que se tendrá acceso al código fuente, se podrá realizar modificaciones en su funcionamiento, si así lo requiriese.

PHP por ser diseñado para trabajar en la web, brindará todas las facilidades para los desarrollos, incluyendo acceso a bases de datos, manejo de archivos, imágenes, web services, y demás. Se puede encontrar una extensa lista de librerías que están a disposición para poder utilizarlas y reducir el tiempo de desarrollo del proyecto.

PHP es compatible con una gran cantidad de sistemas operativos, entre los principales y más utilizados se tienen: Linux, Unix, Windows, Mac OS, entre otros. Lo que implica que PHP es totalmente portable.

Desde la versión 5 de PHP, existe soporte de Orientación a Objetos, lo que permite desarrollar aplicaciones con este paradigma de programación, punto necesario para el proyecto, pues se implementará el Patrón de Diseño MVC.

PHP soporta una gran lista de gestores de bases de datos, en particular utilizaremos PostgreSQL.

1.3.4 SERVIDOR WEB APACHE

A pesar de que existen varias opciones de servidores web, se inclinará por la utilización de Apache, por los siguientes motivos:

Apache corre en una gran cantidad de Sistemas Operativos, al igual que PHP.

Apache es una tecnología de código abierto que lleva muchos años en el mercado, por lo que cuenta con una madurez más que aceptable.

El hecho de que apache sea de código abierto, da la seguridad de que se pueda saber exactamente qué es lo que hace, teniendo la posibilidad de poder auditarlo.

Apache es de diseño modular, por lo que permite adaptarlo a las necesidades de acuerdo a las capacidades de hardware o de requerimientos.

Apache trabaja con una gran cantidad de lenguajes de script, entre los principales se tiene a PHP y Perl.

Apache a nivel mundial es el servidor web más utilizado en servicios de hosting.

1.3.5 SCRUMY: HERRAMIENTA PARA GESTIONAR SCRUM

Se ha escogido scrumy por su simplicidad. Permite la gestión de los sprints backlogs mediante el Scrum task board, accesible a través de la web desde cualquier navegador.

1.3.6 APTANA: IDE PARA DESARROLLO EN PHP

Aptana Software es un IDE basado en Eclipse, éste hecho supone una gran ventaja en el desarrollo de la aplicación pues ya ha sido usado por quienes desarrollarán el sistema, por ende la curva de aprendizaje será mínima, favoreciendo así los tiempos planificados para el desarrollo.

Aptana soporta lenguajes como PHP, Python, Ruby, CSS, Ajax, HTML y Adobe AIR. Tiene la posibilidad de incluir complementos para nuevos lenguajes y funcionalidades.

2 CAPÍTULO 2. DESARROLLO DEL SISTEMA

2.1 ESPECIFICACIÓN DE REQUERIMIENTOS

Este apartado parte de la siguiente premisa: “Los requerimientos no son parte del sistema, los requerimientos definen al sistema”.

Las metodologías Scrum y XP proponen a las “Historias de Usuario” como un medio eficaz para documentar los requerimientos. Unificando estas metodologías el primer paso será el describir de forma amplia las historias de usuario como lo propone XP y posteriormente, como propone SCRUM, se desglosarán tales historias a nivel de tareas (dicho desglose será parte del acápite Análisis). Una historia de usuario estará asignada a una determinada iteración y por ende también sus tareas.

Los requerimientos se recolectarán en reuniones con los involucrados, tanto desarrolladores como usuarios del sistema. En Scrum estas reuniones se denominan Daily Meeting^[72].

Como paso previo, es necesario definir los perfiles que formarán parte del sistema. La definición de perfiles resulta de la determinación de actores que tienen interacción con el sistema.

2.1.1 USUARIOS DEL SISTEMA

Los usuarios que tendrán acceso al sistema son:

Administrador: será el encargado de ejecutar y asegurar el correcto funcionamiento del sistema. Como administrador se define también al dueño de la empresa, la funcionalidad de este usuario será extendida, podrá crear, modificar e incluso eliminar: usuarios, empresas proveedoras, productos, categorías. Un usuario administrador, puede a su vez crear otro usuario Administrador con un perfil limitado, por ejemplo, que pueda solo realizar consultas y no pueda añadir productos.

^[69] Agile Alliance. Daily Meeting

Usuario no registrado: aquel que podrá navegar por la gama de productos, observar sus características, pero no podrá efectuar la compra.

Usuario registrado (cliente): aquel que pueda realizar añadir productos al carrito, visualizar el costo de su compra y finalmente comprar.

2.1.2 HISTORIAS DE USUARIO

Una Historia de Usuario es un requerimiento del negocio desde el punto de vista de un usuario. Permite describir de forma sencilla los requisitos funcionales del sistema. La Tabla 2.1 muestra el formato adoptado para las historias de usuario.

Historia de Usuario	
Número:	
Nombre de Historia:	
Prioridad en Negocio:	Usuario:
Descripción:	
Dependencias:	

Tabla 2.1: FORMATO DE HISTORIA DE USUARIO

Número: el numero secuencial de Historia de Usuario.

Nombre de Historia: texto descriptivo que denomina a la Historia de Usuario.

Prioridad en el Negocio: escala cualitativa (alta/media/baja) que determina la importancia de la implementación de la Historia de Usuario.

Descripción: el detalle propio de la Historia de Usuario. EL requisito funcional del sistema que se desea documentar.

Dependencias: el número de Historia de Usuario de la cual depende, si es el caso.

2.1.3 DEFINICION DE LAS HISTORIAS DE USUARIO

Exceptuando la historia número uno, en cuyo caso los actores son tanto los usuarios como el administrador, las siguientes historias hacen referencia al back end del sistema, es decir, las tareas que involucran al administrador del mismo.

La Tabla 2.2 muestra la Historia de Usuario Ingresar Nuevo Usuario.

Historia de Usuario	
Número: 1	
Nombre de Historia: Ingresar Nuevo Usuario	
Prioridad en Negocio: Baja	Usuario: Administrador
Dependencias:	
Descripción: el administrador podrá crear usuarios del sistema. Cuando se ingresan los datos de un usuario que no es administrador, el usuario pasa a conocerse como cliente.	

Tabla 2.2: Historia de Usuario Ingresar Nuevo Usuario

La Tabla 2.3 muestra la Historia de Usuario Modificar Usuario Existente.

Historia de Usuario	
Número: 2	
Nombre de Historia: Modificar Usuario Existente	
Prioridad en Negocio: Baja	Usuario: Administrador
Dependencias: Historia 1	
Descripción: el administrador podrá modificar usuarios existentes en el sistema. Podrá modificar los datos personales asociados a este usuario, también los privilegios que tiene, así como su estado actual en el sistema, es decir, si consta como usuario habilitado o inhabilitado.	

Tabla 2.3: Historia de Usuario Modificar Usuario

La Tabla 2.4 muestra la Historia de Usuario Eliminar Usuario.

Historia de Usuario	
Número: 3	
Nombre de Historia: Eliminar Usuario Existente	
Prioridad en Negocio: Baja	Usuario: Administrador
Dependencias: Historia 1	
Descripción: el administrador podrá eliminar usuarios existentes en el sistema si así lo requiero	

Tabla 2.4: Historia de Usuario Eliminar Usuario

La Tabla 2.5 muestra la Historia de Usuario Implementación de Perfiles.

Historia de Usuario	
Número: 4	
Nombre de Historia: Implementación de Perfiles	
Prioridad en Negocio: Baja	Usuario: Administrador
Dependencias: Historia de Usuario 1.	
Descripción: el administrador del sistema podrá de determinar si un usuario que accederá a la información de sistema tendrá permisos de lectura, de escritura o una combinación de ellos, y agrupar estos niveles de acceso en su correspondiente perfil. Como consecuencia de esta asignación cada usuario que ingrese al sistema lo hará con sus correspondientes credenciales de acceso.	

Tabla 2.5: Historia de Usuario Implementación de Perfiles

La Tabla 2.6 muestra la Historia de Usuario Ingresar Nuevas Categorías.

Historia de Usuario	
Número: 5	
Nombre de Historia: Ingresar Nuevas Categorías	
Prioridad en Negocio: Alta	Usuario: Administrador
Dependencias:	
Descripción: el administrador podrá ingresar nuevas categorías de productos. Al tratarse de una tienda de artículos de cuero, las categorías hasta ahora pensadas son: bolsos, zapatos y chaquetas, sin embargo no existirá restricción en el ingreso de nuevas categorías.	

Tabla 2.6: Historia de Usuario Ingresar Nuevas Categorías

La Tabla 2.7 muestra la Historia de Usuario Modificar Categorías.

Historia de Usuario	
Número: 6	
Nombre de Historia: Modificar Categorías.	
Prioridad en Negocio: Alta	Usuario: Administrador
Dependencias: Historia de Usuario 5	
Descripción: el administrador podrá modificar las categorías de productos existentes, cambiar su nombre o cambiar su descripción.	

Tabla 2.7: Historia de Usuario Modificar Categorías

La Tabla 2.8 muestra la Historia de Usuario Eliminar Categorías.

Historia de Usuario	
Número: 7	
Nombre de Historia: Eliminar Categorías	
Prioridad en Negocio: Alta	Usuario: Administrador
Dependencias: Historia de Usuario 5	
Descripción: el administrador podrá eliminar las categorías de productos existentes en caso de ya no ser necesarias.	

Tabla 2.8: Historia de Usuario Eliminar Categorías

La Tabla 2.9 muestra la Historia de Usuario Ingresar Nuevos Productos.

Historia de Usuario	
Número: 8	
Nombre de Historia: Ingresar Nuevos Productos	
Prioridad en Negocio: Alta	Usuario: Administrador
Dependencias:	
Descripción: el administrador podrá ingresar nuevos productos, lo que incluye de modo general añadir una imagen y una descripción, y de manera más específica añadir los datos propios del productos, los referentes a su cuantía y disponibilidad. Debe existir la posibilidad de asociar o no el producto a un fabricante.	

Tabla 2.9: Historia de Usuario Ingresar Nuevos Productos

La Tabla 2.10 muestra la Historia de Usuario Modificar Productos

Historia de Usuario	
Número: 9	
Nombre de Historia: Modificar Productos	
Prioridad en Negocio: Alta	Usuario: Administrador
Dependencias: Historia de Usuario 8	
Descripción: el administrador podrá modificar los productos existentes, lo que incluye modificar una imagen y su descripción, su cuantía y disponibilidad, también su fabricante.	

Tabla 2.10: Historia de Usuario Modificar Productos

La Tabla 2.11 muestra la Historia de Usuario Eliminar Productos.

Historia de Usuario	
Número: 10	
Nombre de Historia: Eliminar Productos	
Prioridad en Negocio: Alta	Usuario: Administrador
Dependencias: Historia de Usuario 8	
Descripción: el administrador podrá eliminar los productos existentes, lo que incluye eliminar una imagen y su descripción.	

Tabla 2.11: Historia de Usuario Eliminar Productos

La Tabla 2.12 muestra la Historia de Usuario Ingresar Fabricantes.

Historia de Usuario	
Número: 11	
Nombre de Historia: Ingresar Fabricantes	
Prioridad en Negocio: Alta	Usuario: Administrador
Dependencias:	
Descripción: el administrador podrá ingresar nuevos Fabricantes, lo que incluye añadir un nombre y una descripción. Éstos fabricantes luego podrán ser asociados a los productos que se registren en el sistema.	

Tabla 2.12: Historia de Usuario Ingresar Empresas Proveedoras

La Tabla 2.13 muestra la Historia de Usuario Modificar Fabricantes.

Historia de Usuario	
Número: 12	
Nombre de Historia: Modificar Fabricantes	
Prioridad en Negocio: Media	Usuario: Administrador
Dependencias: Historia de Usuario 11	
Descripción: el administrador podrá modificar las empresas proveedoras, lo que incluye modificar un nombre y una descripción.	

Tabla 2.13: Historia de Usuario Modificar Empresas Proveedoras

La Tabla 2.14 muestra la Historia de Usuario Eliminar Fabricantes.

Historia de Usuario	
Número: 13	
Nombre de Historia: Eliminar Fabricantes	
Prioridad en Negocio: Media	Usuario: Administrador
Dependencias: Historia de Usuario 11	
Descripción: el administrador podrá eliminar las empresas proveedoras en el caso de que ya no trabaje con ellas o que hayan desaparecido del mercado, lo que incluye eliminar un nombre y una descripción.	

Tabla 2.14: Historia de Usuario Eliminar Empresas Proveedoras

La Tabla 2.15 muestra la Historia de Usuario Establecer formas de pago y métodos de envío.

Historia de Usuario	
Número: 14	
Nombre de Historia: Establecer Formas de Pago y métodos de envío	
Prioridad en Negocio: Alta	Usuario: Administrador
Dependencias:	
Descripción: el administrador podrá determinar las formas de pago que estarán disponibles en el sistema.	
Observaciones: escoger los productos, comprarlos y pagar por ellos se ha formulado como una sola Historia de Usuario ya que son actividades transaccionales, dependen la una de la otra.	

Tabla 2.15: Establecer formas de pago

La Tabla 2.16 muestra la Historia de Usuario Gestión de pedidos, compras y pagos.

Historia de Usuario	
Número: 15	
Nombre de Historia: Gestión de pedidos, compras y pagos	
Prioridad en Negocio: Alta	Usuario: Administrador
Dependencias: Historia de Usuario 14	
Descripción: el administrador podrá dar seguimiento a las compras realizadas en la tienda virtual. Para ello necesito visualizar una lista de las compras realizadas, y el estado de éstas (pendientes, en proceso de entrega, entregadas, canceladas, etc). En las compras el administrador tendrá total acceso a los pedidos realizados, eliminar los pedidos según lo requiera. Debe existir la confirmación del pago que un cliente ha realizado por su compra, y para este fin se registrará un comentario en los cambios de estado del pedido.	

Tabla 2.16: Historia de Usuario Gestión de pedidos, compras y pagos

La Tabla 2.17 muestra la Historia de Usuario Manejo de Stock

Historia de Usuario	
Número: 16	
Nombre de Historia: Manejo de Stock	
Prioridad en Negocio: Media	Usuario: Administrador
Dependencias:	
Descripción: el administrador podrá programar notificaciones para la gestión del stock para saber a tiempo cuándo un producto está por terminarse.	

Tabla 2.17: Historia de Usuario Manejo de Stock

La Tabla 2.18 muestra la Historia de Usuario Información de Ventas

Historia de Usuario	
Número: 17	
Nombre de Historia: Información de Ventas	
Prioridad en Negocio: Media	Usuario: Administrador
Dependencias:	
Descripción: El propietario del negocio podrá observar un resumen de los productos más vendidos, de las últimas ventas ejecutadas y de los productos vendidos en un periodo de tiempo determinados, así podrá adelantarse a las necesidades del negocio.	

Tabla 2.18: Historia de Usuario Informacion de Ventas

La Tabla 2.19 muestra la Historia de Usuario Ingresar Información de la Empresa.

Historia de Usuario	
Número: 18	
Nombre de Historia: Ingresar Información de la Empresa	
Prioridad en Negocio: Baja	Usuario: Administrador
Dependencias:	
Descripción: el administrador podrá ingresar datos de la empresa como nombre, contactos, ubicación e incluso una imagen descriptiva.	

Tabla 2.19: Historia de Usuario Ingresar Información de la Empresa

La Tabla 2.20 muestra la Historia de Usuario Modificar Información de la Empresa.

Historia de Usuario	
Número: 19	
Nombre de Historia: Modificar Información de la Empresa	
Prioridad en Negocio: Baja	Usuario: Administrador
Dependencias: Historia de Usuario 18	
Descripción: el administrador podrá modificar datos de la empresa como su nombre, contactos, ubicación e incluso una imagen descriptiva.	
Observaciones:	

Tabla 2.20: Historia de Usuario Modificar Información de la Empresa

La Tabla 2.21 muestra la Historia de Usuario Eliminar Información de la Empresa.

Historia de Usuario	
Número: 20	
Nombre de Historia: Eliminar Información de la Empresa	
Prioridad en Negocio: Baja	Usuario: Administrador
Dependencias: Historia de Usuario 18	
Descripción: el administrador podrá eliminar datos de la empresa como su nombre, contactos, ubicación e incluso una imagen descriptiva.	

Tabla 2.21: Historia de Usuario Eliminar Información de la Empresa

Las siguientes historias de usuario son definidas para el Front-End, es decir, involucran la participación de los usuarios del sistema que no son administradores.

La Tabla 2.22 muestra la Historia de Usuario Crear Cuenta de Usuario

Historia de Usuario	
Número:21	
Nombre de Historia: Crear Cuenta de usuario	
Prioridad en Negocio: Media	Usuario: Usuario no registrado
Dependencias:	
Descripción: un usuario del sistema podrá ingresar los datos acorde a su perfil de usuario (dirección, teléfono, etc).	

Tabla 2.22: Historia de Usuario Crear Cuenta de Usuario

La Tabla 2.23 muestra la Historia de Usuario Modificar Cuenta de Usuario

Historia de Usuario	
Número: 22	
Nombre de Historia: Modificar Cuenta de usuario	
Prioridad en Negocio: Baja	Usuario: Usuario registrado
Dependencias: Historia 21.	
Descripción: un usuario del sistema podrá realizar modificaciones o actualizaciones de su perfil (dirección, teléfono, etc.).	

Tabla 2.23: Historia de Usuario Modificar Cuenta de Usuario

La Tabla 2.24 muestra la Historia de Usuario Eliminar Cuenta de Usuario

Historia de Usuario	
Número: 23	
Nombre de Historia: Eliminar Cuenta de usuario	
Prioridad en Negocio: Baja	Usuario: Usuario registrado – Administrador
Dependencias: Historia 21.	
Descripción: un usuario del sistema podrá darse de baja. El mecanismo consistirá en presionar un botón o enlace que enviará una notificación al administrador y sólo este es quien puede dar de baja al usuario.	

Tabla 2.24: Historia de Usuario Eliminar Cuenta de Usuario

La Tabla 2.25 muestra la Historia de Usuario Buscar un Producto.

Historia de Usuario	
Número: 24	
Nombre de Historia: Buscar un producto	
Prioridad en Negocio: Alta	Usuario: Usuario registrado y no registrado
Dependencias:	
Descripción: un usuario del sistema podrá observar los productos, su descripción, sus características, buscar lo que necesita antes de aceptar crear una cuenta de usuario.	

Tabla 2.25: Historia de Usuario Buscar un Producto

La Tabla 2.26 muestra la Historia de Usuario Anadir un producto a la compra.

Historia de Usuario	
Número: 25	
Nombre de Historia: Añadir un producto a la compra	
Prioridad en Negocio: Alta	Usuario: Usuario registrado
Dependencias: Historia 1.	
Descripción: un usuario del sistema podrá escoger los productos que deseo adquirir y la cantidad que necesito poner en stock, además de datos adicionales que definan al producto.	

Tabla 2.26: Historia de Usuario Ingresar Nuevo Usuario

La Tabla 2.27 muestra la Historia de Usuario Eliminar un producto antes de realizar la compra

Historia de Usuario	
Número: 26	
Nombre de Historia: Eliminar un producto antes de realizar la compra	
Prioridad en Negocio: Alta	Usuario: Usuario registrado
Dependencias: Historia 1.	
Descripción: un usuario del sistema podrá retractarse de ordenar algún producto que estuvo en el listado inicial de opciones.	

Tabla 2.27: Historia de Usuario Eliminar un producto antes de realizar la compra

La Tabla 2.28 muestra la Historia de Usuario Realizar una Compra

Historia de Usuario	
Número: 27	
Nombre de Historia: Realizar una compra.	
Prioridad en Negocio:	Usuario: Usuario registrado
Dependencias:	
<p>Descripción: Muestra en detalle lo añadido en el carro de compras, en este módulo se podrá modificar la cantidad de productos, eliminar productos, o regresar al catálogo de productos. Una vez que el usuario desee comprar, deberá presionar en el botón comprar y se mostrará una pantalla con el método de pago (transferencia bancaria – en el correo de confirmación al usuario deberá llegarle información detallada como número de cuenta, etc.; Contrareembolso). Además del método de envío, el cual puede llegar a tener un valor adicional. Todo este proceso debería ser notificado vía correo al usuario (confirmación de compra, cambio de estado del pedido etc.).</p>	

Tabla 2.28: Historia de Usuario Realizar una compra

2.2 ANÁLISIS

En esta etapa se determinan los tiempos y el esfuerzo necesarios para el desarrollo del sistema. Dicho de otra manera, se formaliza la planificación del proyecto.

Al gestionar el proyecto con SCRUM los documentos que se proponen a este respecto son:

- PRODUCT BACKLOG.
- SPRINT BACKLOG.

2.2.1 PRODUCT BACKLOG

El Product Backlog es un listado priorizado de funcionalidades o requisitos que son básicamente los objetivos que el cliente tiene como expectativa del producto de software a desarrollarse.^[51]

El listado de historias de usuario será jerarquizado en virtud de su importancia para el negocio, dicha importancia se calcula multiplicando la urgencia por el impacto.

La urgencia se medirá en una escala del 1 al 5 siendo 1 la de menor urgencia. El impacto se medirá en una escala del 1 al 5 siendo 5 la de mayor impacto negativo si llegara a no concretarse.

Adicionalmente el Product Backlog estima a groso modo el tiempo de implementación de las tareas (medido en días).

Una vez definida la importancia de las tareas se asignan cada una de ellas a una iteración. La iteración más próxima incluirá a las tareas de mayor importancia.

El Product Backlog describirá lo que se pretende que el producto de software haga, y lo que respecta a cómo solventar esa funcionalidad ya corre por cuenta

^[51] PICHLER , Roman. Agile Product Management with Scrum.

del Equipo. Este Product Backlog constantemente se irá actualizando en cada reunión de planificación a medida de que avance el proyecto hasta la culminación del producto de software.

El formato del Product Backlog se muestra en la Tabla 2.29.

	PILA DEL PRODUCTO				
ID	HISTORIA DE USUARIO	URGENCIA	IMPACTO	IMPORTANCIA	TIEMPO ESTIMADO (días)

Tabla 2.29: Formato Del Product Backlog

La descripción de los campos de la Tabla 30, es la siguiente:

ID: es el número de la Historia, nos permitirá localizarla aun si esta cambia de nombre.

Historia de Usuario: nombre de la Historia.

Urgencia: menor urgencia 1. Mayor urgencia 5.

Impacto: menor impacto 1. Mayor impacto 5.

Importancia: urgencia x Impacto

Tiempo Estimado: tiempo en llevar a cabo la tarea medido en días.

2.2.1.1 Definición Del Product Backlog

La tabla 2.30 muestra el listado de las historias de usuario del sistema que constituyen la Pila del Producto o Product Backlog .

2.2.2 SPRINT BACKLOG Y BURN DOWN CHART

El desglose del PRODUCT BACKLOG permite puntualizar las tareas a realizar durante una iteración. Estas tareas se resumen en los SPRINT BACKLOG. Un SPRINT BACKLOG describe la manera cómo el equipo va a implementar los requisitos durante el sprint. Las tareas deben ser lo más concretas posibles, llegando a un nivel de detalle que al traducirlas en tiempo no duren más de 16 horas, caso contrario, dicha tarea deberá dividirse en otras de menor envergadura.

Una particularidad de las tareas de un sprint es que no se asignan a un miembro del equipo, es el participante quien la toma con total libertad según le convenga.

El Sprint Backlog es por lo general visualizado mediante tableros físicos, los cuales son denominados Scrum Task Board, que siempre es visible para todos, y en el que se listan las tareas que se van a realizar en cada Sprint, y conforme se vaya avanzando en el desarrollo, estas tareas cambiarán de estado mientras se las mueve hacia la derecha del tablero (pendiente, en curso, para verificar y terminado), de manera que se pueda ver en qué tareas se está trabajando y en cuales se observa retraso.

La figura 2.1 muestra el formato general del Scrum Task Board. Para ejemplificar hemos colocado algunas de las tareas de este proyecto usando la herramienta en línea llamada scrummy a la que se puede acceder mediante el enlace <http://scrummy.com>

De modo general y como se observa en la figura se puede decir que el tablero está constituido por historias de usuarios, posteriormente divididas en tareas. Estas tareas son tomadas por el miembro del equipo que considera estar en la aptitud de desarrollarla, se identifican con color diferente por cada miembro del equipo involucrado. Dichas tareas ocupan una posición en el tablero de acuerdo al estado en el que se encuentran.

Con algo más de detalle podemos mencionar que las secciones del tablero se distribuyen de la siguiente manera:

STORIES: aquí se enlistan las historias de usuario que forman parte de un sprint.

TO DO: aquí se detallan las tareas necesarias para concluir la historia de usuario. Inicialmente todas las tareas están en estado TO DO, posteriormente serán movidas a la columna de avance correspondiente.

IN PROGRESS: aquí se colocan las tareas que se están desarrollando.

VERIFY: a este estado pasan las tareas que terminaron de desarrollarse conforme al requerimiento establecido y que deberán ser probadas.

DONE: aquí se colocan las tareas que tras probarlas han sido aceptadas como válidas, es decir, son funcionalmente correctas y no necesitan corrección.

PRODUCT BACKLOG					
ID	HISTORIA DE USUARIO	URGENCIA	IMPACTO	IMPORTANCIA	TIEMPO ESTIMADO (días)
1	Ingresar Nuevo Usuario	5	4	20	4
2	Modificar Usuario	4	4	16	4
3	Eliminar Usuario	4	4	16	4
4	Implementación de Perfiles	5	4	20	4
5	Ingresar Categorías	4	3	12	4
6	Modificar Categorías	4	3	12	4
7	Eliminar Categorías	3	2	12	4
8	Ingresar Productos	4	3	12	3
9	Modificar Productos	3	2	12	3
10	Eliminar Productos	4	3	9	3
11	Ingresar Empresas Proveedoras	3	3	9	3
12	Modificar Empresas Proveedoras	3	3	9	3
13	Eliminar Empresas Proveedoras	3	3	9	3
14	Establecer Formas de Pago y Métodos de Envío	3	3	9	4
15	Gestión Pedidos, Compras, Pagos	2	4	8	3
16	Manejo de Stock	2	4	8	2
17	Generar Información de Ventas	3	2	6	2
18	Ingresar Información Empresa	2	2	4	2
19	Modificar Información de la Empresa	2	4	4	2
20	Eliminar Información de la Empresa	2	2	4	2
21	Crear Cuentas de Usuario	5	4	20	2
22	Modificar Cuentas de Usuario	4	4	16	2
23	Eliminar Cuentas de Usuario	4	4	16	2
24	Buscar Producto	5	4	25	6
25	Añadir Producto a la Compra	5	5	25	6
26	Eliminar Producto	5	5	25	6
27	Realizar la Compra	5	5	25	6

Tabla 2.30: Product Backlog

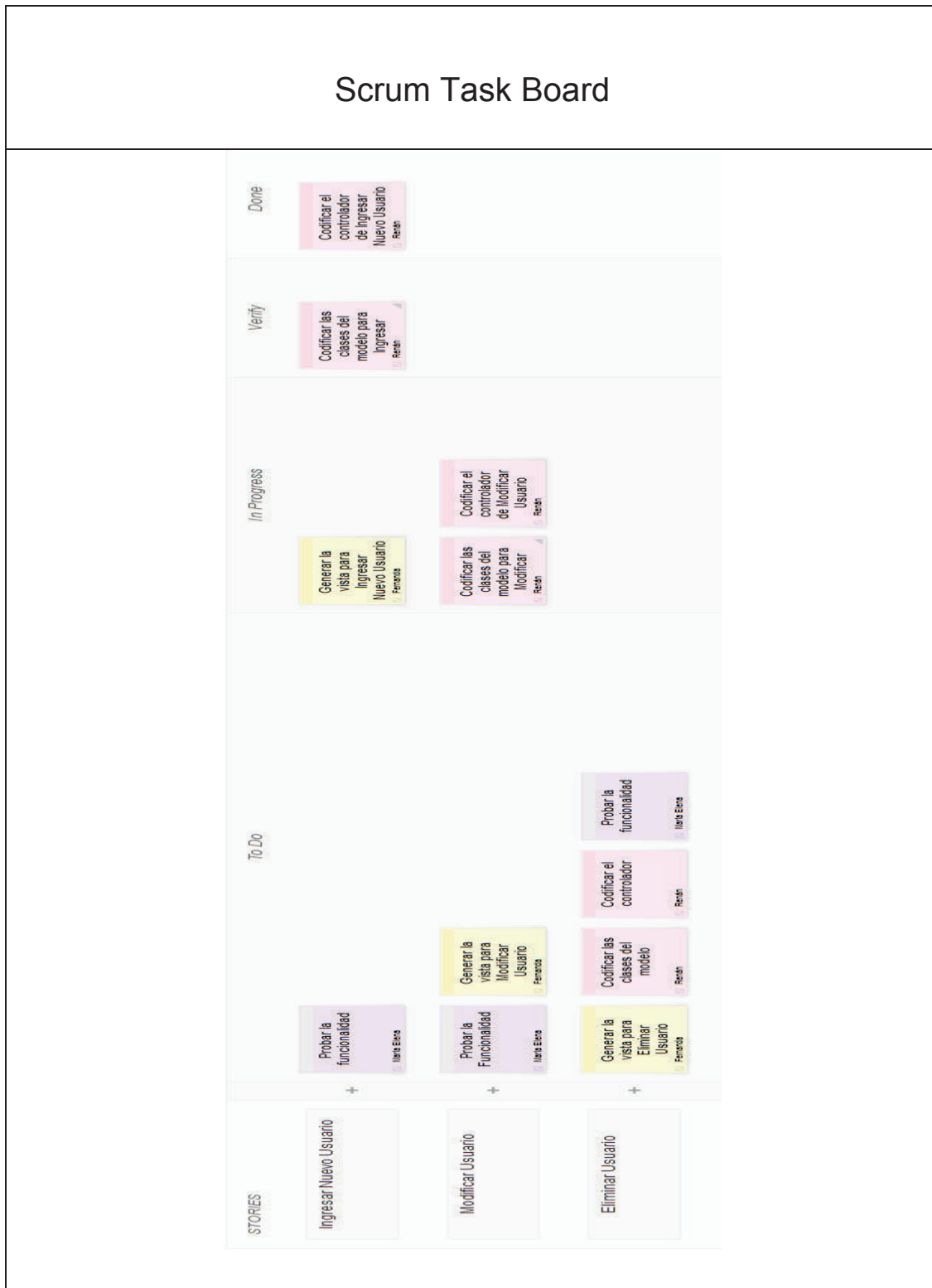


Figura 2.1: Scrum Task Board realizado usando la herramienta en línea Scrumy

La organización de las tareas en el tablero permite una rápida visualización del estado ACTUAL de las tareas. Esta representación es por tanto útil durante el proceso ágil de desarrollo, careciendo de sentido que forme parte de la documentación posterior.

Por su parte el Sprint Backlog, que será básicamente el desglose de las tareas puntuales programadas para un sprint, formará parte de los documentos del proyecto.

La tabla 2.31 muestra el formato adoptado para la definición del Sprint Backlog.

HISTORIA DE USUARIO	TAREAS	TIEMPO (días)	EJECUTADA POR:

Tabla 2.31: Formato Del Sprint Baklog

Los campos de la tabla siguiente son:

Historia de Usuario: nombre de la Historia de Usuario asignada al Sprint.

Tareas: desglose de la Historia de Usuario a nivel de tareas de corta duración.

Tiempo: tiempo estimado en días para la culminación de la tarea.

Ejecutada por: nombre de la persona que, conforme a los principios de Scrum, toma voluntariamente la tarea en concordancia con sus puntos fuertes como equipo.

El **BURNDOWN CHART** es la demostración grafica de la cantidad de tareas pendientes al iniciar cada Sprint. Permite visualizar el progreso del proyecto. Como en toda metodología de desarrollo ágil, está permitido añadir nuevos requisitos, estos también se reflejan en la gráfica, así como las modificaciones que surjan en el trayecto.

2.2.3 PLANIFICACIÓN INICIAL DE LAS ITERACIONES/SPRINTS

Como resultado de la priorización de Historias de Usuario en el Product Backlog , se puede realizar una planificación tentativa de las iteraciones necesarias en este ciclo de desarrollo de software.

La tabla 2.32 muestra una primera aproximación, la que podríamos llamar ideal, del Product Backlog dividido en sprints o iteraciones. Los sprints inicialmente asignados podrían ser susceptibles de modificación, como es propio en desarrollos de tipo ágil, de ocurrir estos cambios serán explicados puntualmente en el desarrollo de cada Sprint.

Como la metodología que se está usando en este proyecto es de tipo iterativo e incremental se hace necesario un ajuste en la planificación en cada siguiente iteración. Claramente el siguiente resultado dependerá del anterior. Si los tiempos estimados son los correctos, no habrá modificaciones, caso contrario deberán ser aumentados o disminuidos dependiendo el caso.

2.3 DISEÑO

2.3.1 DESCRIPCIÓN DEL PATRÓN DE ARQUITECTURA

El patrón de diseño adoptado es MVC (Modelo-Vista-Controlador). Este patrón separa el modelo o lógica del negocio de su presentación o vista, enlazándolos mediante el controlador^[101].

Con algo más de detalle podemos decir que el Modelo envuelve el comportamiento y los datos del dominio de aplicación; la Vista despliega la información y el Controlador interpreta las entradas generadas por el mouse o el teclado de usuario, informando al modelo cuando es necesario un cambio de estado. En la Figura 2.2, se muestra el Ciclo de vida o funcionamiento de MVC.

^[101] MARSTON, Tony. The Model-View-Controller (MVC) Design Pattern for PHP.

ID	HISTORIA DE USUARIO	SPRINT PROGRAMADO
24	Añadir Producto a la Compra	1
25	Eliminar Producto	1
26	Realizar la Compra	1
20	Crear Cuentas de Usuario	1
21	Modificar Cuentas de Usuario	2
22	Eliminar Cuentas de Usuario	2
1	Ingresar Nuevo Usuario	2
2	Modificar Usuario	2
3	Eliminar Usuario	2
4	Implementación de Perfiles	2
5	Ingresar Categorías	3
6	Modificar Categorías	3
7	Eliminar Categorías	3
8	Ingresar Productos	3
9	Modificar Productos	4
10	Eliminar Productos	4
11	Ingresar Empresas Proveedoras	4
12	Modificar Empresas Proveedoras	4
13	Eliminar Empresas Proveedoras	4
14	Gestión Pedidos, Compras, Pagos	5
15	Manejo de Stock	5
16	Generar Información de Ventas	5
17	Ingresar Información Empresa	5
18	Modificar Información de la Empresa	5
19	Eliminar Información de la Empresa	5

Tabla 2.32: DIVISIÓN IDEAL DEL PRODUCT BACKLOG EN SPRINTS (ITERACIONES)

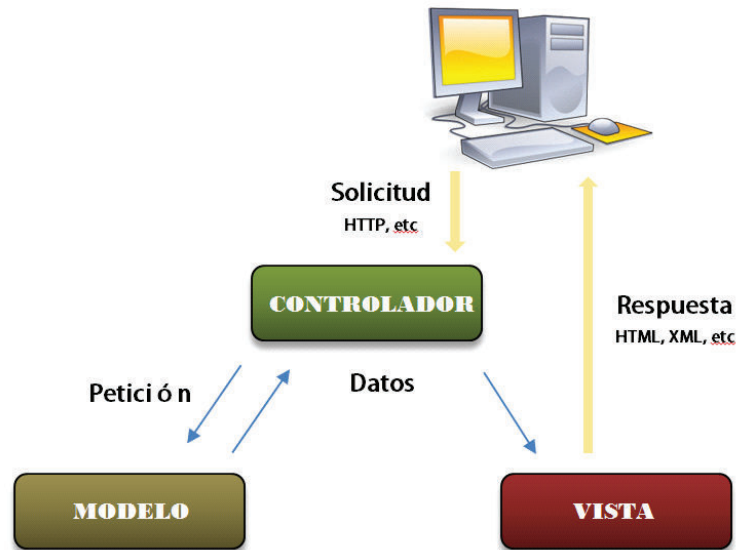


Figura 2.2: Funcionamiento Del Patrón De Arquitectura Modelo-Vista-Controlador

2.3.2 MODELO DE BASE DE DATOS

La figura 2.6 muestra el modelo entidad relación generado para el sistema ESIWECOM. La herramienta usada para el diagrama fue pgModeler, de uso libre, que ha sido diseñada para el uso específico de PostgreSQL, generando el código resultante para la creación de las tablas.

2.3.3 MODELO DE NAVEGABILIDAD

2.3.3.1 Navegabilidad Para El Administrador

La figura 2.3 muestra el diagrama de navegabilidad para el administrador del sistema. Vale recordar que este usuario tiene acceso a toda la funcionalidad del sistema, por ende, se muestra la navegabilidad exclusiva para él, es decir, la que forma parte del backend.

2.3.3.2 Navegabilidad Para El Usuario No Registrado

La figura 2.4 muestra la navegabilidad para el usuario no registrado. Este usuario puede visitar el sitio, observar la oferta de productos en el menú PRINCIPAL, pero no puede realizar una compra, por ello al intentar acceder a los menús LOGIN, CUENTA, CARRITO y COMPRA el resultado será un mensaje que le indicará que esas opciones son para usuarios registrados.

2.3.3.3 Navegabilidad Para El Usuario Registrado

La figura 2.5 muestra la navegabilidad para el usuario registrado. Este usuario puede visitar el sitio, observar la oferta de productos en el menú HOME y tras ingresar su usuario y contraseña en el menú LOGIN, puede añadir productos al carrito y finalmente realizar una compra.

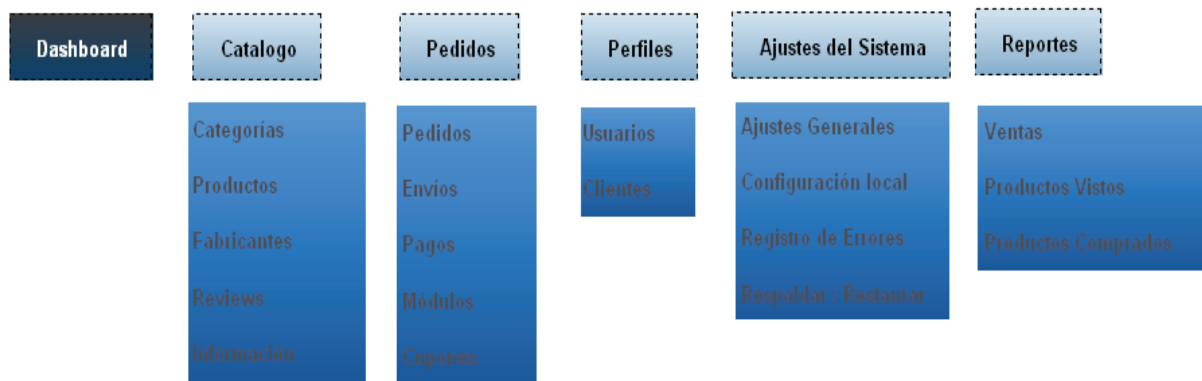


Figura 2.3: Diagrama De Navegabilidad En El Backend



Figura 2.4: . Diagrama De Navegabilidad Para El Usuario No Registrado

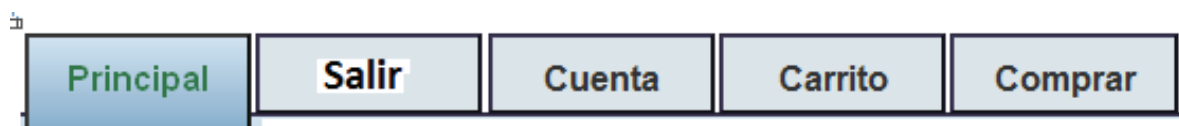


Figura 2.5: Diagrama De Navegabilidad Para El Usuario Registrado

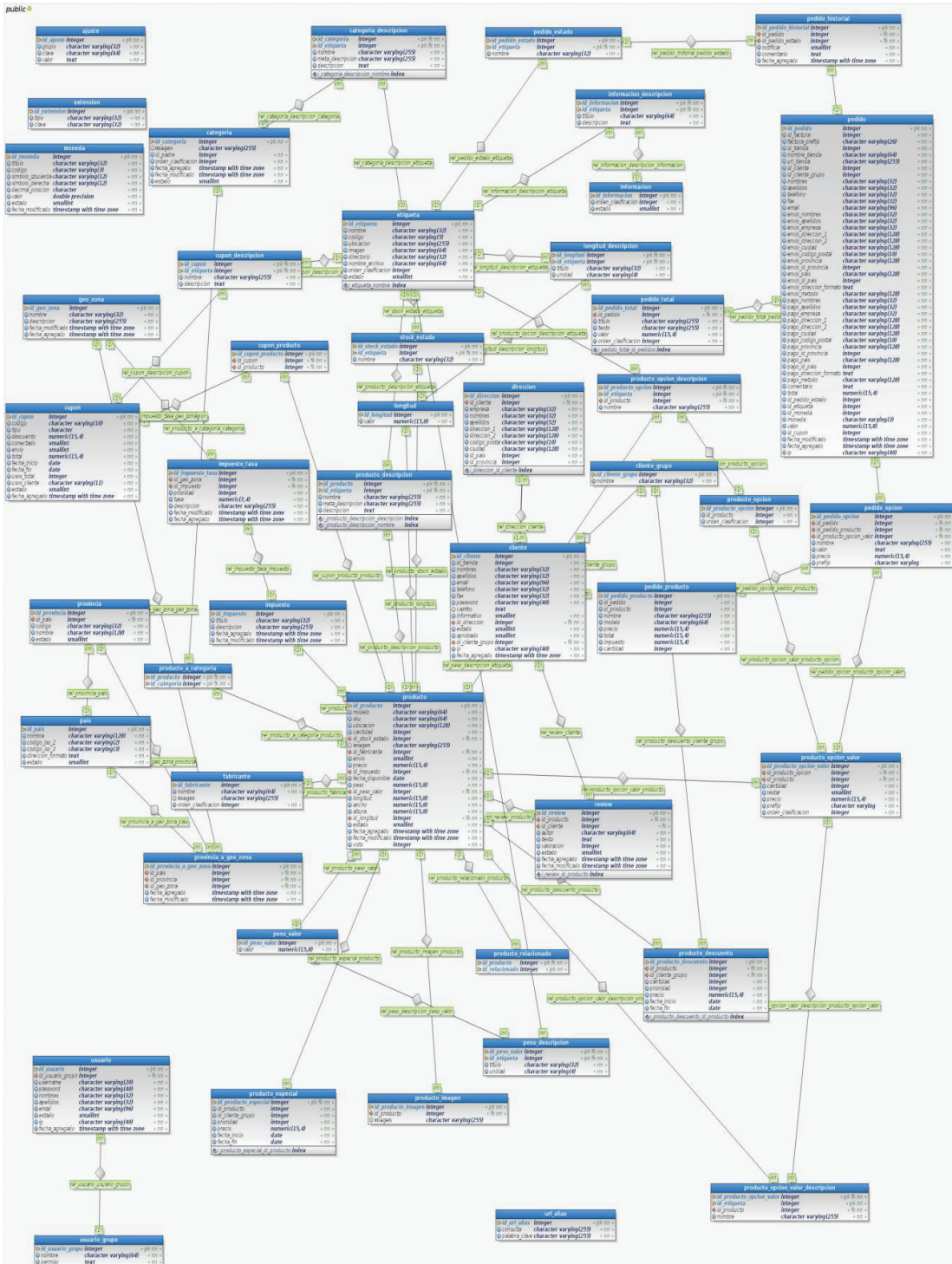


Figura 2.6: Modelo Entidad Relación Del Sistema ESIWECOM

2.4 IMPLEMENTACIÓN

La implementación del proyecto se realiza en PHP usando el IDE APTANA.

- Para crear el proyecto se debe seleccionar File – New – Web Project. En la siguiente pantalla se dará la ubicación del archivo y el nombre que se le asignará.
- Una vez elegido el nombre y ubicación se da click en Next obteniendo una ventana con varios tipos de opciones, html y dos tipos de html5. Si se presiona finish sin escoger una de estas opciones se acepta la opción por defecto (html).
- Ya creado el proyecto, el siguiente paso es crear las páginas dentro de él.

2.4.1 ESTÁNDAR DE CODIFICACIÓN

Mantener estándares de codificación permite dar formato a los archivos PHP que se generarán durante toda la implementación del sistema, adicionalmente facilitará la posterior tarea de mantenimiento del sistema. También permitirá trabajar con orden y evitar problemas el momento de ensamblar los módulos que han sido desarrollados independientemente por cada miembro del equipo.

Para estandarizar el código se ha decidido tomar como referencia algunas sugerencias del estándar de codificación Zend^[102].

- Identación de 4 espacios, no tabulación.
- La longitud recomendable para una línea de código es de 80 caracteres. El tamaño máximo de cualquier línea de código PHP es de 120 caracteres.
- Los nombres de clases pueden contener sólo caracteres alfanuméricos. Los números están permitidos en los nombres de clase, pero desaconsejados en la mayoría de casos. Las barras bajas (_) están permitidas solo como

^[102] Zend Framework para PHP. Estándares de Codificación..

separador de ruta. Si el nombre de una clase está compuesto por más de una palabra, la primera letra de cada palabra debe aparecer en mayúsculas.

- Los nombres de funciones pueden contener únicamente caracteres alfanuméricos. Los guiones bajos (_) no están permitidos. Los números están permitidos en los nombres de función pero no se aconseja en la mayoría de los casos. Los nombres de funciones deben empezar siempre con una letra minúscula.
- Para las variables de instancia que son declaradas con el modificador "private" o "protected", el primer carácter de la variable debe ser una única barra baja (_). Este es el único caso admisible de una barra baja en el nombre de una variable. Las variables declaradas como "public" no pueden empezar nunca por barra baja.
- Las constantes pueden contener tanto caracteres alfanuméricos como barras bajas (_). Los números están permitidos. Todas las letras pertenecientes al nombre de una constante deben aparecer en mayúsculas.

2.4.2 IMPLEMENTACIÓN ACORDE AL PATRÓN MVC (MODELO VISTA CONTROLADOR)

La implementación del sistema define claramente dos escenarios, el front-end y el back-end. El front-end será para los usuarios registrados y no registrados. El back-end será el dominio del usuario administrador. En la estructura de directorios asociada tanto al front-end como al back-end se encuentran tres carpetas: modelo, vista, controlador, que albergarán el código alusivo a su nombre y que permitirán dar funcionalidad al sistema en base al patrón MVC. Cabe recordar que en la implementación se ha adoptado la notación de clases y variables que propone el estándar de codificación Zend (descrito en el acápite previo).

2.4.2.1 Estructura de directorios del sistema

El sistema está desarrollado con el patrón de diseño MVC (Modelo – Vista - Controlador) que permite la separación entre los datos y la presentación.

Se definió una estructura de directorios que ayuda a separar la parte de la administración del sistema (Back-End) en su correspondiente directorio “admin”, y los archivos relacionados con la interface que será utilizado y mostrado a los

usuarios clientes (Front-End), identificado por el directorio “catalogo”, para conseguir independencia entre ambas partes.

La Figura 2.7 muestra una visión general de los elementos del sistema.

Tanto en la parte Front-End así como en la parte Back-End se utiliza MVC, por cuanto, ambas partes pudieran convivir en el mismo servidor o en su defecto, cada parte en servidores distintos, apuntando a una misma base de datos.

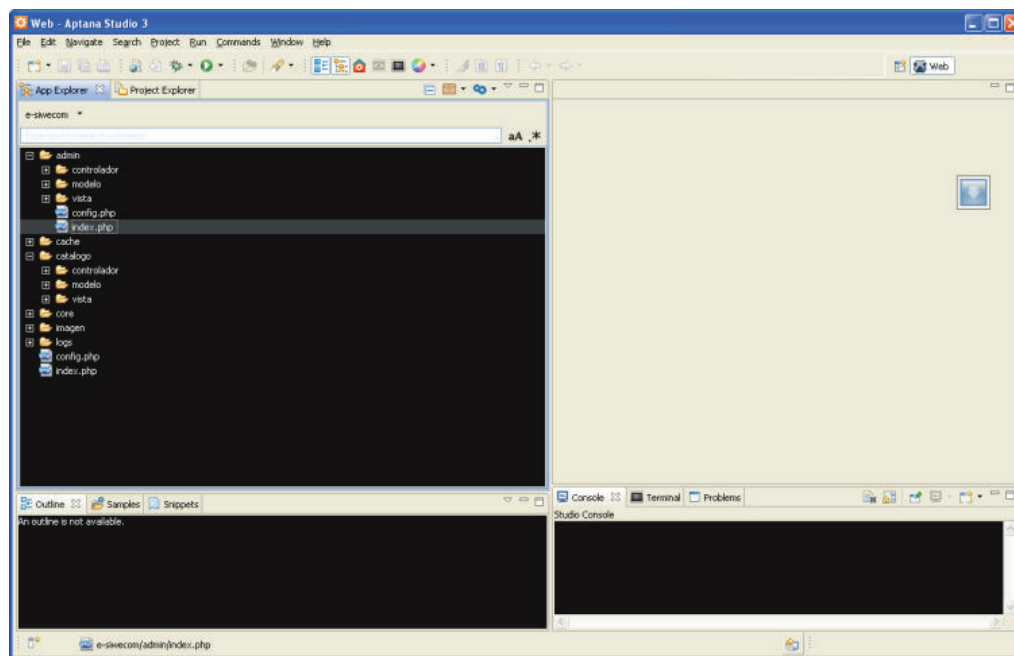


Figura 2.7: Visión General de los Elementos del Sistema en Aptana

Tanto Front-End (catalogo), así como el Back-End (admin) poseen la misma estructura de directorios:

Controlador: es responsable de manejar la lógica de la aplicación, interactuando con el modelo.

Este directorio contiene los distintos módulos o funcionalidades del sistema ordenados por subdirectorios, en los que se encuentra las respectivas clases que los define.

Modelo: es responsable de acceder a la base de datos de forma directa y realizar las operaciones pertinentes con la misma, respecto a las peticiones emitidas por el controlador.

De forma similar, se encuentran ordenadas las clases de los modelos en subdirectorios.

Vista: es responsable de mostrar la información solicitada por el modelo, en un formato adecuado para que el usuario pueda interactuar con la aplicación y sus datos.

Encontramos varios subdirectorios dentro de este directorio, en los que se localizan archivos que definen la vista (imágenes, estilos, javascript) y el directorio “plantilla” dentro del cual se ubican más subdirectorios y que tienen correspondencia con los subdirectorios del controlador, por lo que la estructura en ambos casos es la misma. Además, cada archivo dentro de este subdirectorio son archivos con código PHP embebido en HTML.

Adicionalmente a los directorios “admin” y “catalogo”, tenemos otros directorios que cumplen una función importante dentro del sistema, entre los cuales tenemos:

Core: este directorio contiene clases y funciones utilizadas por el Front-End y el Back-End del sistema. Dentro de este directorio encontramos el subdirectorio “conectordb” donde ubicamos la clase conectora de la base de datos (postgres), el subdirectorio “motor” donde se encuentran las clases que definen a controladores, modelos y otras partes del sistema; y por último tenemos al subdirectorio “librerias”, que básicamente contiene las clases y funciones usadas por controladores y modelos de todo el sistema tanto en Front-End así como en el Back-End.

La figura 2.8 nos da una visión de cómo se organizan los elementos que forman parte del Core del sistema.

Imagen: este directorio contiene todas las imágenes que son subidas por el administrador de imágenes de la parte del Back-End. Estas imágenes corresponden a productos, imágenes adicionales y versiones almacenadas de imágenes que han sido cambiadas de tamaño por el propio sistema.

Logs: en este directorio se almacena un archivo de texto plano (log) en donde se guardan los errores que el sistema arroje en su funcionamiento.

Cache: al generar las páginas web de forma dinámica, el sistema genera archivos que contienen información de consultas realizadas previamente, las cuales ayudan a acceder a esas páginas o información ya consultada, de forma más rápida, es decir, se produce un almacenamiento en caché de esas páginas que probablemente se reitere su visita o consulta.

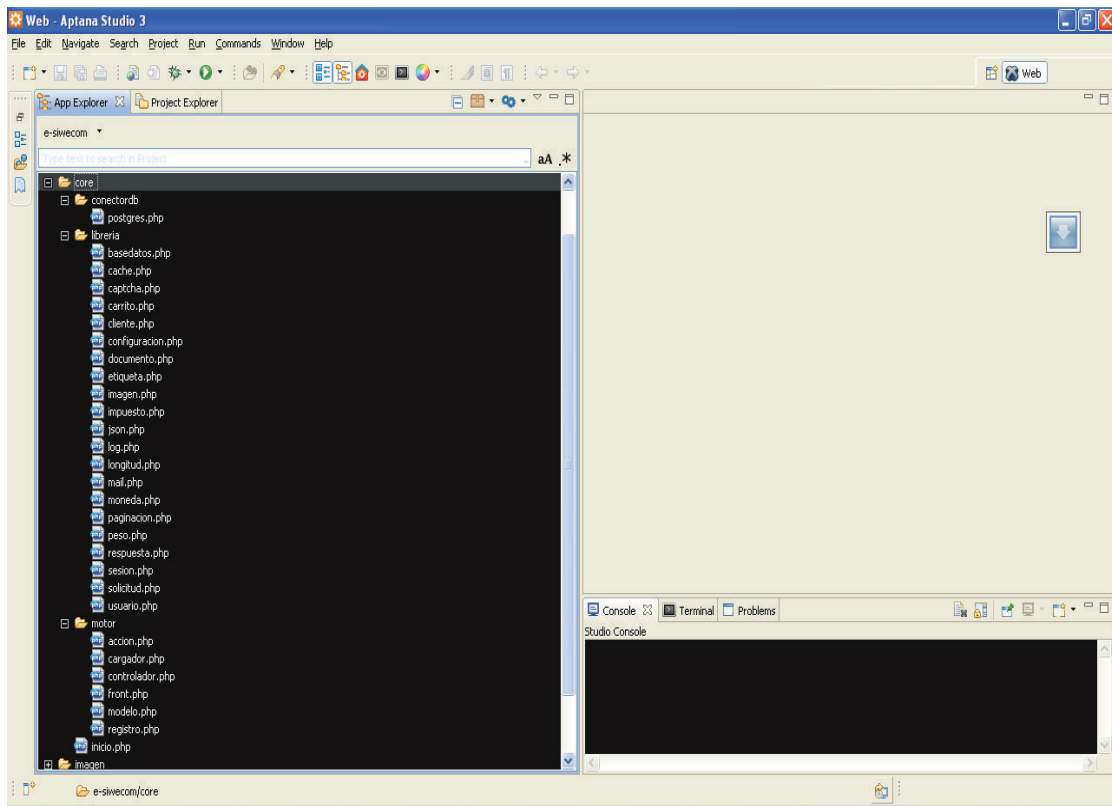


Figura 2.8: Visión de los Directorios que forman el Core del sistema

Dentro de este directorio se puede observar archivos que se van generando conforme se utilice el sistema, nombrados inicialmente con la palabra “cache” seguido del nombre de las clases que son accedidas o utilizadas e IDs con secuencias numéricas de estas consultas almacenadas en caché.

2.4.3 EJECUCIÓN REAL DE LAS ITERACIONES/SPRINTS

En este apartado se definirán el sprint backlog y el Burn down chart resultante de cada etapa de análisis ocurrido durante el desarrollo de proyecto. Vale mencionar que de forma general lo primero en llevarse a cabo es el cuerpo del sistema, en sí la opción dada al usuario de buscar productos y realizar una compra, originalmente los datos que necesitará el usuario para la navegación son ingresados a la base de datos de forma directa, sin ninguna interfaz. Las interfaces de mantenimiento, es decir, el back-end del sistema resultaron ser de menor urgencia por eso se implementan en sprints posteriores.

Antes de entrar en el detalle de cada Sprint, es necesario hacer algunas precisiones a fin de entender cómo empata lo que se presentará con la metodología usada,

- 1) Las tareas definidas en cada Sprint (y que en cada caso se detallan en una tabla) fueron seleccionadas en una reunión de equipo denominada *Sprint Planning Meeting*. Para esta reunión los miembros del equipo estuvieron físicamente concentrados en un mismo lugar por un tiempo de aproximadamente 4 horas.
- 2) Ya en el desarrollo del sprint, se realizó diariamente una reunión de corta duración (alrededor de 10 minutos) para dar seguimiento al progreso del trabajo. Estas reuniones, denominadas *Daily Meeting*, fueron realizadas en su mayoría gracias a la ayuda de la tecnología, en ocasiones vía telefónica y en su mayoría mediante conversaciones tipo texto.
- 3) Como resultado del sprint se presenta un burndown chart, la explicación de dicho gráfico básicamente constituye el *Sprint Review*, pues permite llegar a un punto de y verificación de las funcionalidades que se completaron..
- 4) Finalmente, se cierra el Sprint revisando la priorización de las tareas del Product Backlog por si algún cambio es necesario. En la metodología, esta revisión de historias de usuario se denomina *Product Backlog Refinement* O *Product Backlog Grooming* (Replanificación Del Proyecto).

Estos 4 puntos aquí precisados fueron descritos con más detalle en el capítulo 1 como parte del subtema 1.2.4.3 (Actividades De Scrum) y constituyen una importante característica de la metodología que son las diferentes reuniones de equipo, cada una con su propósito particular. Si bien Scrum no determina un formato de acta de reunión, es más, ni siquiera las exige y en la mayor parte de la literatura ni siquiera se la propone, dichas actas pueden resultar de mucha utilidad, más aún al tratarse de equipos de trabajo numerosos en los que fácilmente puede olvidarse quién dijo qué y a qué acuerdos se llegó. En el desarrollo de este proyecto no fue necesaria la formalización de un acta de reunión, sin embargo, proponemos un formato que podría ser útil para equipos de desarrollo que trabajen físicamente en un mismo lugar y que, por su experiencia en lo que respecta a llegar a acuerdos, les resulte necesario tales documentos (Véase Anexo I).

Pese a no evidenciarlo en actas, el ya mencionado proceso fue realizado cada una de las veces que la metodología lo propone, es decir, una vez por cada iteración. El resultado de tal proceso se refleja en la descripción de cada sprint y los consiguientes objetivos en él alcanzados.

2.4.3.1 Sprint 1 o Primera Iteración

Tanto el Sprint Backlog como el Burn down Chart presentados como resultado de esta iteración, corresponden a espacios de tiempo reales, no ideales.

Sprint Backlog De La Primera Iteración (Sprint 1).

La Tabla 2.33 recoge las historias de usuario que forman parte del Sprint 1, determinando las tareas que lo conforman, la duración de las mismas y registrando el responsable de cumplirla.

Como se observa en la tabla 2.33 se ha sombreado con un color diferente la tarea denominada Probar la funcionalidad, ya que si bien aparece desglosada en 4 líneas diferentes, al momento de su ejecución se convierte en solo una, pues la ejecutará la Sra. María Elena, propietaria del negocio, y las pruebas de la

funcionalidad de dichos componentes se harán en un solo día (este dato es importante al momento de contar las tareas para graficar el trabajo pendiente).

La duración de cada tarea (medida en horas) es asignada por la persona que se responsabilizó de ella, la apreciación del tiempo en llevarla a cabo responde básicamente al factor experiencia.

HISTORIA DE USUARIO	TAREAS	TIEMPO (horas)	TAREA TOMADA POR:
Buscar Producto	Generar la vista	12	Fernanda
	Codificar el modelo	7	Renán
	Codificar el controlador	7	Renán
	Probar la funcionalidad	1	María Elena
Añadir Producto a la Compra	Generar la vista	12	Fernanda
	Codificar el modelo	7	Renán
	Codificar el controlador	7	Renán
	Probar la funcionalidad	1	María Elena
Eliminar Producto de la Compra	Generar la vista	12	Fernanda
	Codificar el modelo	7	Renán
	Codificar el controlador	7	Renán
	Probar la funcionalidad	1	María Elena
Crear cuenta de Usuario	Generar la vista	12	Fernanda
	Codificar el modelo	7	Renán
	Codificar el controlador	7	Renán
	Probar la funcionalidad	1	María Elena
	Total de horas a invertir	108	
	Total de tareas	13	

Tabla 2.33: SPRINT BACKLOG 1

Al ser el tiempo a invertir un total de 108 horas y tomando en cuenta hay dos desarrolladores trabajando 4 horas por día (es decir 8 horas efectivas de trabajo), con una simple división del total de horas para las horas efectivas de trabajo (108/8) llegamos a la conclusión que el sprint debe tener una duración de 13,5

días. Este valor está apenas dentro de lo aceptable según Scrum, pues la planificación de cada Sprint no debe exceder de 2 semanas.

Burn down Chart Del Sprint 1

En este Sprint, al cual en base a una perspectiva optimista se le ha asignado una duración de 13 días, se considera un equipo dedicado al desarrollo de 2 miembros con un trabajo ideal de 4 horas al día.

La figura 2.9 muestra el contraste de la pendiente de trabajo ideal versus la real. Como se observa Al inicio del sprint 1 se contabilizan 13 tareas por realizar, pero en términos reales 3 no llegan a culminarse en el tiempo establecido, esto nos indica que el siguiente sprint debe calcularse con más cuidado, asignando menos tareas, o en su defecto, aumentando más días al Sprint.

La diferencia entre la pendiente de trabajo real y la ideal que evidencia la figura puede ser el indicativo de una estimación inicial de tiempo demasiado optimista. Debido a ello, al día de conclusión del sprint, quedaron 3 tareas pendientes.

La funcionalidad que quedó inconclusa en este sprint deberá desarrollarse en el siguiente sprint.

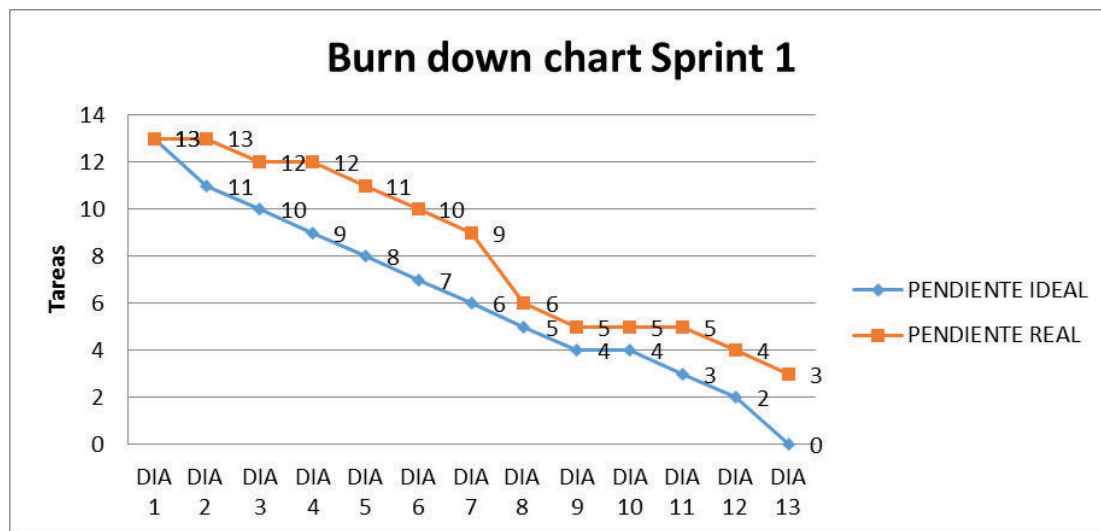


Figura 2.9: Burn down chart Sprint 1

Resultado del Sprint 1 o Primera iteración

La figura 2.10 permite observar la creación de la página principal que muestra al usuario final los productos que podrá comprar. La interfaz de usuario no registrado es un módulo prácticamente funcional, como se observa en la división superior derecha ya se pueden escoger productos y añadirlos al carrito de compra. La figura 2.11 muestra que puede realizarse la búsqueda avanzada de productos. La figura 2.12 muestra que el menú carrito ya despliega la lista de artículos añadidos a la compra. La figura 2.13 indica la funcionalidad ya existente de eliminar un producto antes de realizar la compra.

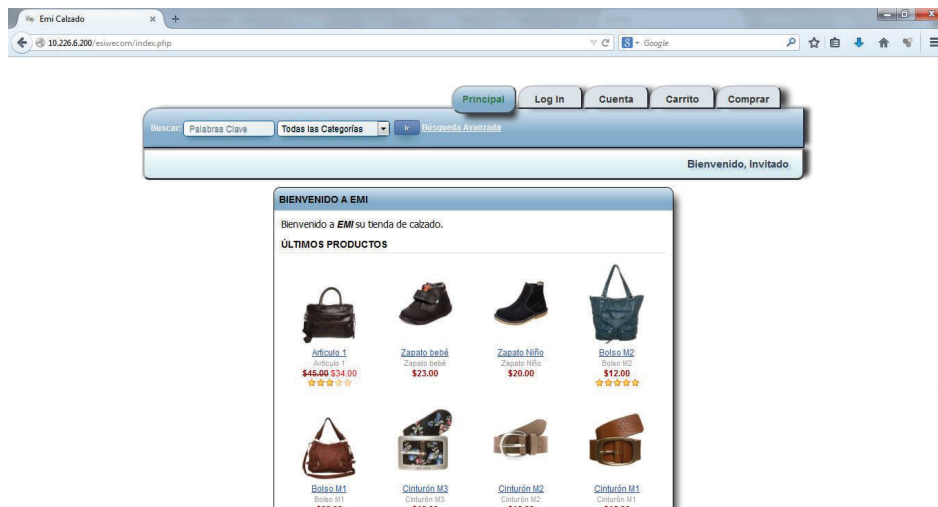


Figura 2.10: Buscar Producto

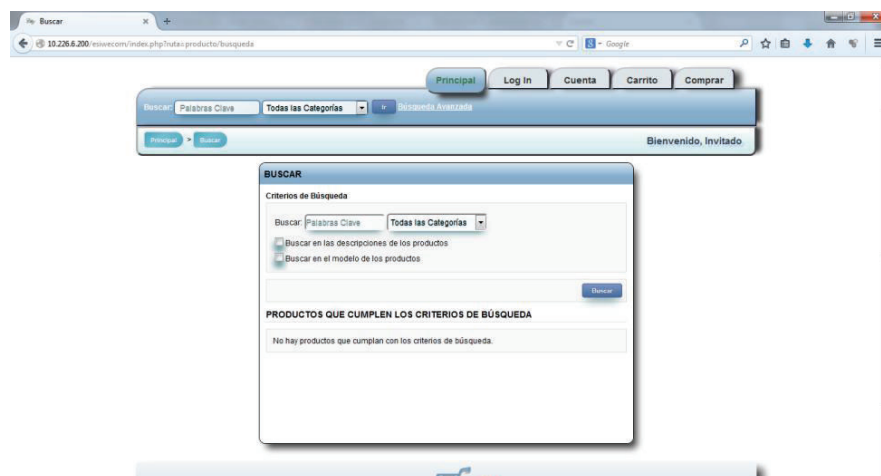


Figura 2.11: Búsqueda Avanzada

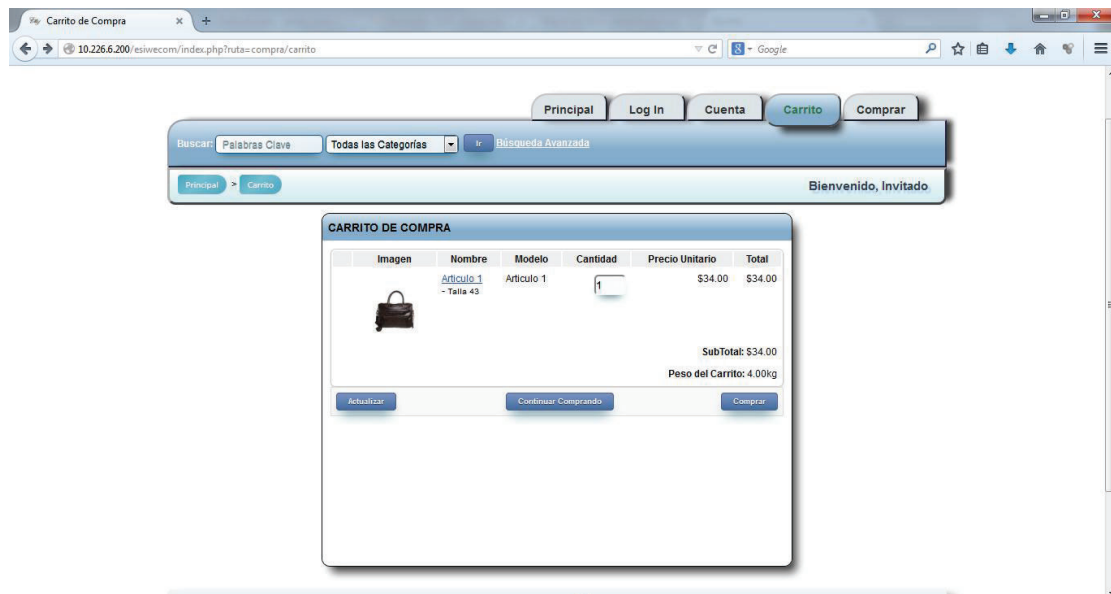


Figura 2.12: Añadir Producto al Carrito de Compra

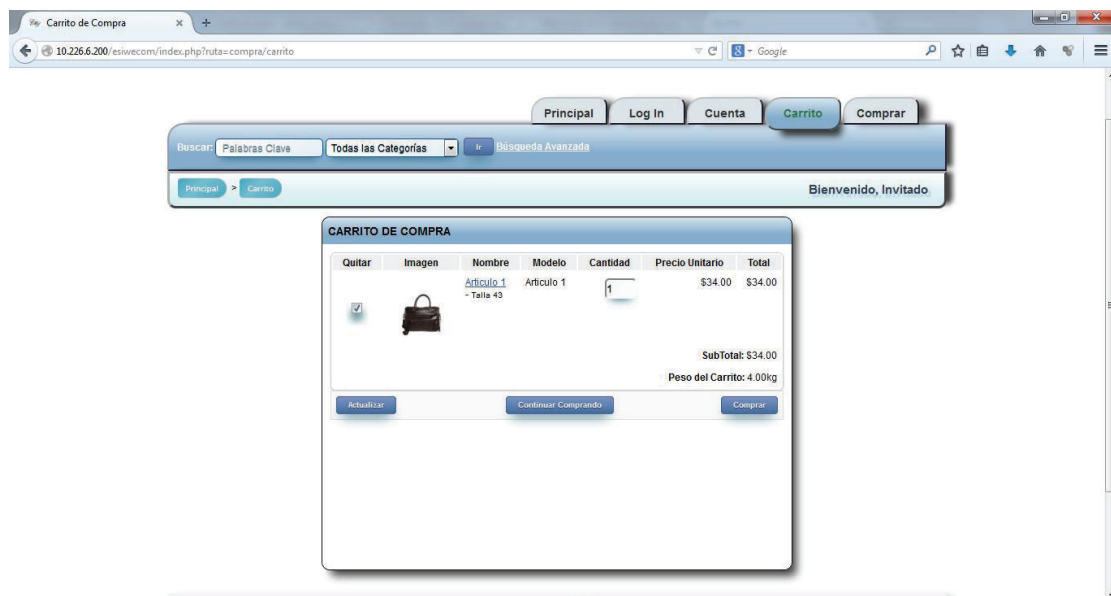


Figura 2.13: Eliminar Producto de la Compra

La vista de la tarea crear cuenta de usuario está ya implementada, sin embargo aún está inconclusa la funcionalidad, esto se evidencia en la figura 2.14. Cabe recalcar que bajo la lógica con la que se concibió el sistema, solo un usuario

registrado, es decir un cliente, puede finalizar el proceso de compra. La figura 2.15 resumen la interfaz principal resultado del Sprint 1.

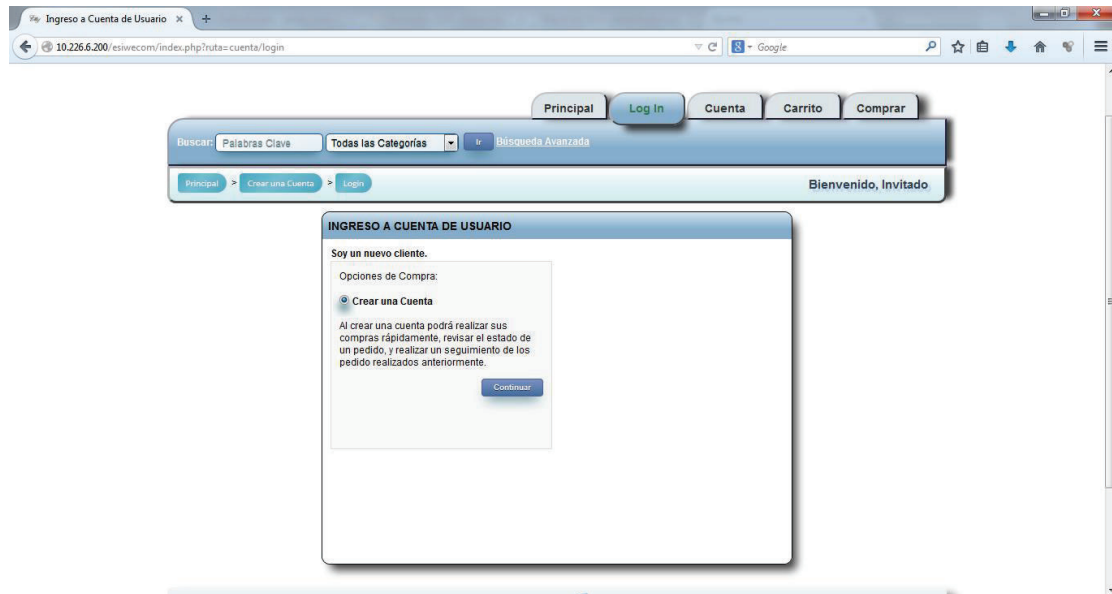


Figura 2.14: Crear cuenta de Usuario

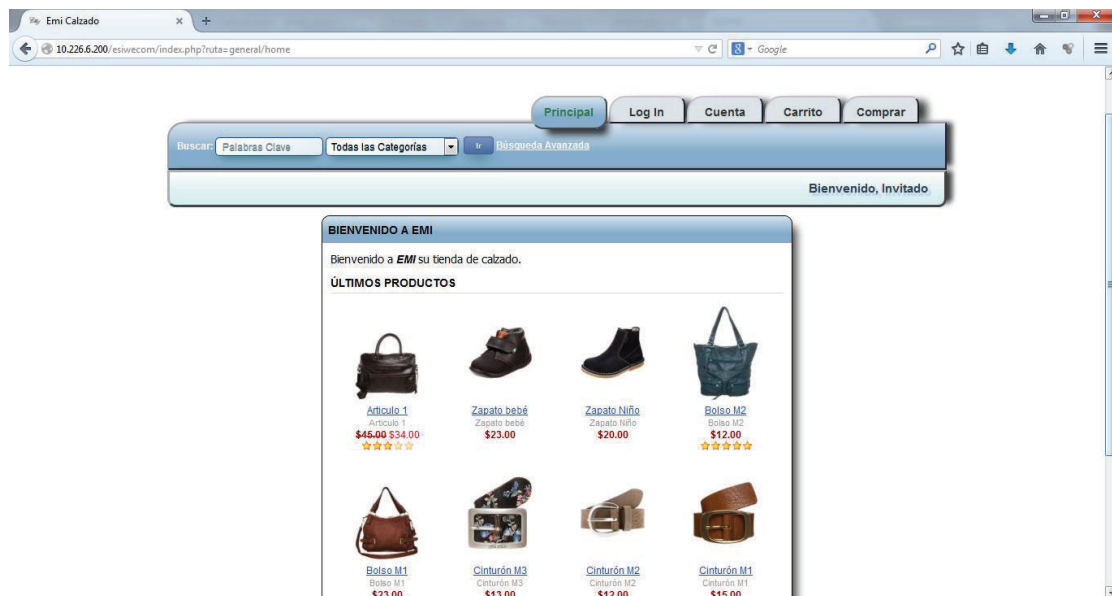


Figura 2.15: Interfaz resultante de la primera iteración (Sprint 1)

2.4.3.2 Sprint 2 o Segunda Iteración

A continuación se muestran los resultados de la planificación del segundo sprint.

Sprint Backlog De La Segunda Iteración (Sprint 2).

Las primeras tareas en este Sprint son aquellas que quedaron inconclusas en el Sprint previo (estas tareas se citan en letra cursiva). La Tabla 2.34 muestra las historias de usuario que forman parte del Sprint 2, sus correspondientes tareas, la duración de las mismas y el responsable de cumplirla.

HISTORIA DE USUARIO	TAREAS	TIEMPO (horas)	TAREA TOMADA POR:
<i>Crear cuenta de Usuario</i>	<i>Generar la vista</i>	<i>10</i>	<i>Fernanda</i>
	<i>Codificar el modelo</i>	<i>6</i>	<i>Renán</i>
	<i>Codificar el controlador</i>	<i>6</i>	<i>Renán</i>
	<i>Probar la funcionalidad</i>	<i>1</i>	<i>María Elena</i>
Realizar la Compra	Generar la vista	12	Fernanda
	Codificar el modelo	7	Renán
	Codificar el controlador	7	Renán
	Probar la funcionalidad	1	María Elena
Modificar cuenta de Usuario	Generar la vista	12	Fernanda
	Codificar el modelo	7	Renán
	Codificar el controlador	7	Renán
	Probar la funcionalidad	1	María Elena
Eliminar cuenta de Usuario	Generar la vista	12	Fernanda
	Codificar el modelo	7	Renán
	Codificar el controlador	7	Renán
	Probar la funcionalidad	1	María Elena
	Total de horas a invertir	104	
	Total de tareas	13	

Tabla 2.34: SPRINT BACKLOG 2

En el caso de las tareas que no se cumplieron en el Sprint 1 y se adjuntaron a este, el tiempo necesario para su implementación ha variado con respecto a la apreciación inicial, pues si bien estas tareas no llegaron a término en la primera iteración, si tuvieron un avance.

La suma del tiempo de este sprint da un total de 108 horas. Se cuenta con dos desarrolladores trabajando 4 horas por día (es decir 8 horas efectivas de trabajo). Dividimos el total de horas necesarias para las horas efectivas de trabajo ($108/8$) y obtenemos como resultado que este sprint debe tener una duración de 13 días, sin embargo se le añadirá un día más del estimado en el cálculo a fin de establecer un colchón a favor del desarrollo.

Burn down Chart Del Sprint 2

Al Sprint se le ha asignado una duración de 14 días, se considera un equipo dedicando al desarrollo de 2 miembros con un trabajo ideal de 4 horas al día.

Al igual que en el Sprint 1, en el Sprint 2 se contabilizan 13 tareas. La figura 13 muestra la evolución de las tareas en el tiempo asignado.

Como se observa en la figura 2.16 la pendiente de trabajo real resultó estar bajo la pendiente de trabajo ideal, esto indica que los objetivos se alcanzaron antes del tiempo planificado. La explicación a este particular se da ya que la estimación de las tareas que quedaron inconclusas en el sprint anterior fue dada con cierta amplitud, eso se refleja en la gráfica ya que cuando se esperaba que se queden 12 tareas pendientes, ya solo quedaban 10.

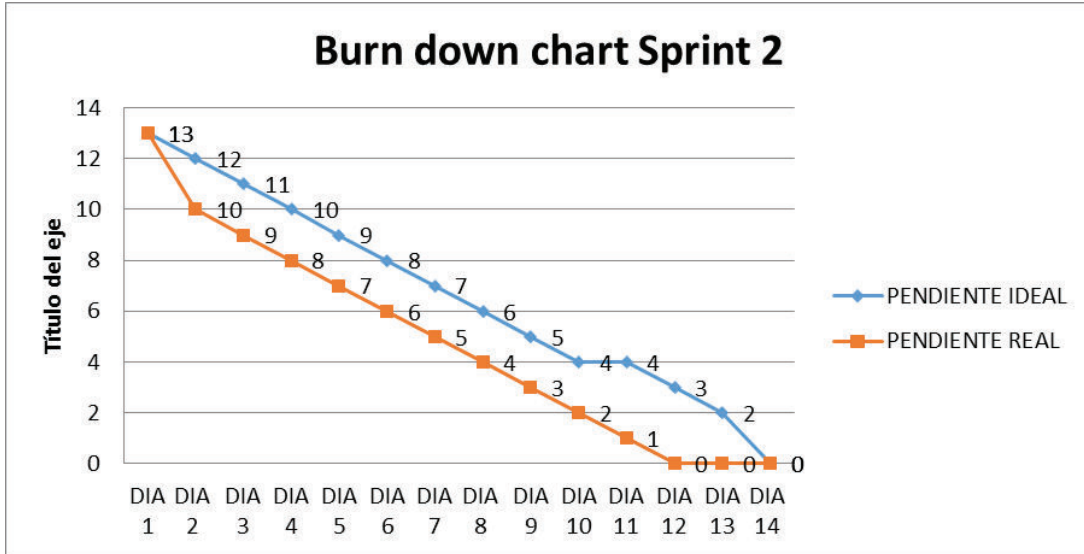


Figura 2.16: Burn down chart Sprint 2

Resultado del Sprint 2 o Segunda iteración

La funcionalidad para la creación de usuarios está ya implementada. La figura 2.17 muestra la creación de la cuenta de usuario desde el front-end del sistema y la figura 2.18 muestra la interfaz correspondiente para esta tarea desde el back-end

Principal Log In Cuenta Carrito Comprar

Buscar: Palabras Clave Todas las Categorías Búsqueda Avanzada

Principal > Cuenta > Registrar Bienvenido, Invitado

CREAR CUENTA

Si ya tiene una cuenta con nosotros, por favor, ingrese a nuestro sistema en la [Página Login](#).

Sus Datos Personales

* Nombre:

* Apellidos:

* E-Mail:

* Teléfono:

Fax:

Su Dirección

Empresa:

* Dirección 1:

Dirección 2:

* Ciudad:

Código Postal:

Figura 2.17: Crear cuenta de Usuario (Front End)

Figura 2.18: Crear cuenta de Usuario (Back End)

La figura 2.19 muestra la implementación de la funcionalidad relacionada a realizar una compra, en esta debe escogerse la forma de envío del producto. La figura 2.20 indica que debe escogerse también la forma de pago al realizar una compra.

Figura 2.19: Realizar la Compra – Información Sobre el Envío

Información del Pago

Principal > Carrito > Envío > Pago

Hola, Renán Guerrero

INFORMACIÓN DEL PAGO

Dirección de Pago

Por favor, escoja la dirección donde desea que se envíe la factura.

[Cambiar Dirección](#)

Dirección de Pago
Renán Guerrero
Carcelén
Quito
Pichincha
Ecuador

Método de Pago

Por favor, seleccione el método de pago deseado para este pedido.

Contra Reembolso
 Transferencia Bancaria

Agregue comentarios a su pedido

[Volver](#) He leído y estoy de acuerdo con [Términos & Condiciones](#) [Continuar](#)

Figura 2.20: Realizar la Compra – Información Sobre el Pago

Como lo muestra la figura 2.21, una vez escogidos el método de pago y envío, se efectiviza la compra y ésta se confirma mediante la emisión de la factura.

Confirmación de Compra

10.226.6.200/esivecom/index.php?ruta=compra/confirmar

Hola, Renán Guerrero

CONFIRMACIÓN DE COMPRA

Método de Envío Serventrega (Peso: 4.00kg) Cambiar	Dirección de Envío Renán Guerrero Carcelén Quito Pichincha Ecuador Cambiar	Dirección de Pago Renán Guerrero Carcelén Quito Pichincha Ecuador Cambiar
---------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------

Producto	Modelo	Cantidad	Precio	Total
Artículo 1 - Talla 43	Artículo 1	1	\$34.00	\$34.00
				SubTotal: \$34.00
				Serventrega (Peso: 4.00kg): \$5.00
				12% IVA: \$4.68
				Total: \$43.68

Ingrese su código, pulse el botón "Aplicar Cupón" y vea que su descuento al instante es aplicado a su pedido.

Cupón: [Aplicar Cupón](#)

Por favor, transfiera el monto total según la siguiente información:

Para realizar el pago de su pedido por favor realizar el depósito en la cuenta corriente número: 12445566 del Banco del Pichincha.

Muchas Gracias por su compra.
Su pedido no se enviará hasta que recibamos el pago.

Figura 2.21: Realizar la Compra – Confirmación

Como parte del sprint 2 también se consiguió dar funcionalidad a la modificación de las cuentas de usuario tanto en el front-end (figura 2.22) como en el back-end (figura 2.23)

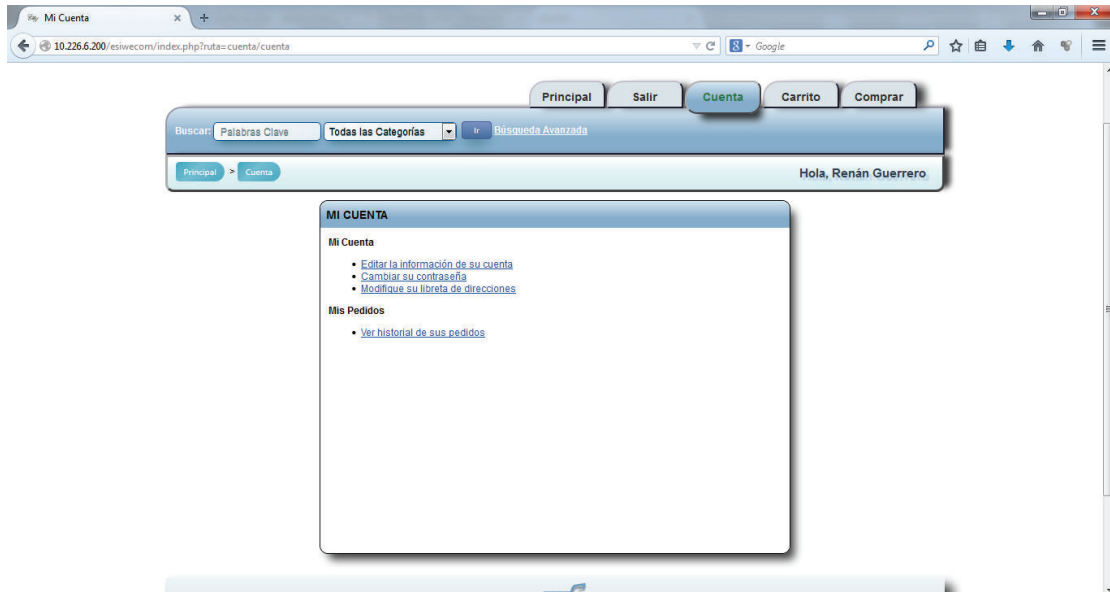


Figura 2.22: Modificar cuenta de Usuario (Front end)

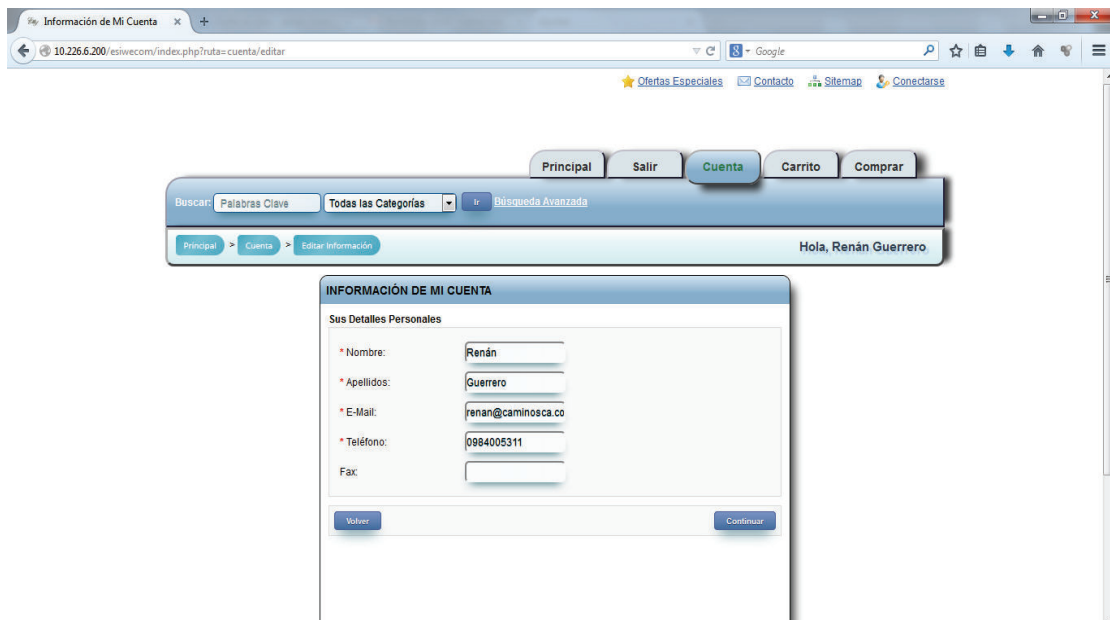


Figura 2.23: Eliminar cuenta de Usuario (En el Backend)

2.4.3.3 Sprint 3 o Tercera Iteración

A continuación se muestran los resultados de la planificación del tercer sprint.

Sprint Backlog De La Tercera Iteración (Sprint 3).

La Tabla 2.35 muestra las historias de usuario que forman parte del Sprint 3, sus correspondientes tareas, la duración de las mismas y el responsable de cumplirla.

La suma del tiempo de este sprint da un total de 62 horas. El tiempo estimado ha disminuido en relación al sprint 2 debido a que el desarrollo de las tareas actuales reutiliza código del sprint anterior.

HISTORIA DE USUARIO	TAREAS	TIEMPO (horas)	TAREA TOMADA POR:
Implementación de Grupos de Usuarios	Generar la vista	10	Fernanda
	Codificar el modelo	3	Renán
	Codificar el controlador	3	Renán
	Probar la funcionalidad	1	María Elena
Crear Usuario	Generar la vista	10	Fernanda
	Codificar el modelo	3	Renán
	Codificar el controlador	3	Renán
	Probar la funcionalidad	1	María Elena
Modificar Usuario	Generar la vista	8	Fernanda
	Codificar el modelo	3	Renán
	Codificar el controlador	3	Renán
	Probar la funcionalidad	1	María Elena
Eliminar Usuario	Generar la vista	6	Fernanda
	Codificar el modelo	3	Renán
	Codificar el controlador	3	Renán
	Probar la funcionalidad	1	María Elena
	Total de horas a invertir	58	
	Total de tareas	13	

Tabla 2.35: SPRINT BACKLOG 3

Se cuenta con dos desarrolladores trabajando 4 horas por día (es decir 8 horas efectivas de trabajo). Dividimos el total de horas necesarias para las horas efectivas de trabajo (62/8) y obtenemos como resultado que este sprint debe tener una duración de 8 días, sin embargo se le añadirá dos días más a fin de establecer un colchón a favor del desarrollo.

Burn down Chart Del Sprint 3

Al Sprint se le ha asignado una duración de 10 días, se considera un equipo dedicando al desarrollo de 2 miembros con un trabajo ideal de 4 horas al día.

En este Sprint se contabilizan 13 tareas. La figura 22.24 muestra la evolución de las tareas en el tiempo asignado.

Como se observa en la figura 2.24 la pendiente de trabajo real se muestra bastante irregular, estando inicialmente por debajo de lo esperado y recuperándose entre los días 4 y 7 para terminar dentro del plazo estimado.

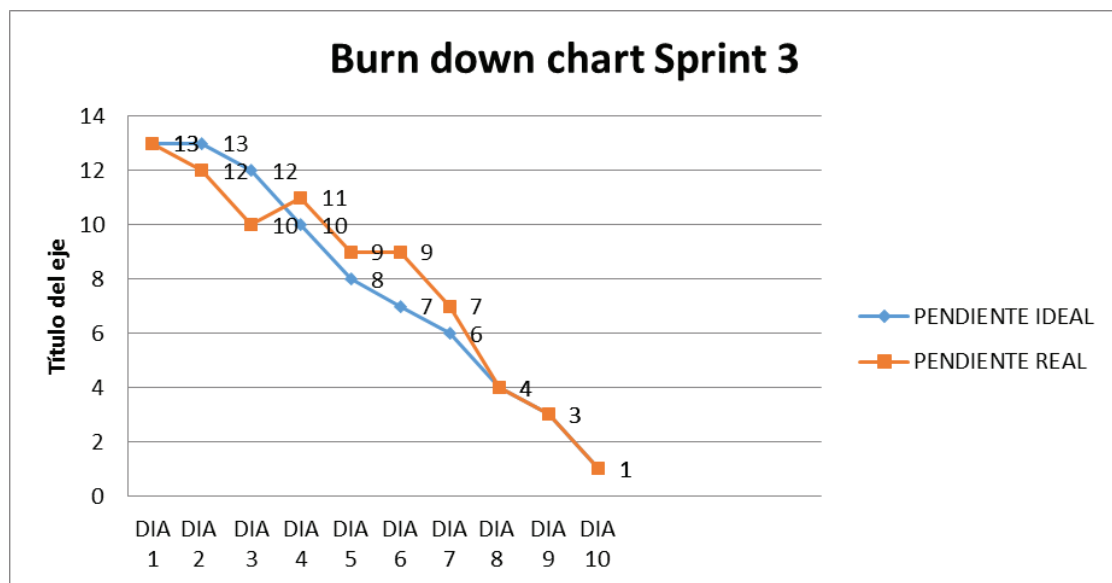


Figura 2.24: Burn down chart Sprint 3

Podemos concluir que el tiempo de cada tarea no fue el preciso, o bien que el factor de eficiencia de trabajo de los desarrolladores no fue constante en el

tiempo. Sin embargo lo importante es que se culminó con el objetivo en el tiempo planteado, pues como muestra la imagen desde el día 8 las tareas pendientes fueron en número iguales a las esperadas, aunque haciendo uso del colchón de tiempo establecido.

Resultado del Sprint 3 o Tercera iteración

El Sprint 2 permitió obtener la funcionalidad de administración de cuentas de usuario para los clientes del sistema, es decir los usuarios no registrados que pasan a registrarse a fin de realizar una compra. En el Sprint 3 se obtiene la funcionalidad de crear usuarios del sistema, administradores del mismo

Como puede verse la decisión agrupar las tareas de creación, modificación y eliminación de usuarios y clientes del sistema en un solo módulo que se corresponde con el menú Perfiles. El submenú usuarios hace referencia a quienes tendrán acceso al back-end del sistema, usuarios grupos permite establecer un perfil para dicho usuario, asignando los permisos que el usuario requiera (Figura 2.25).

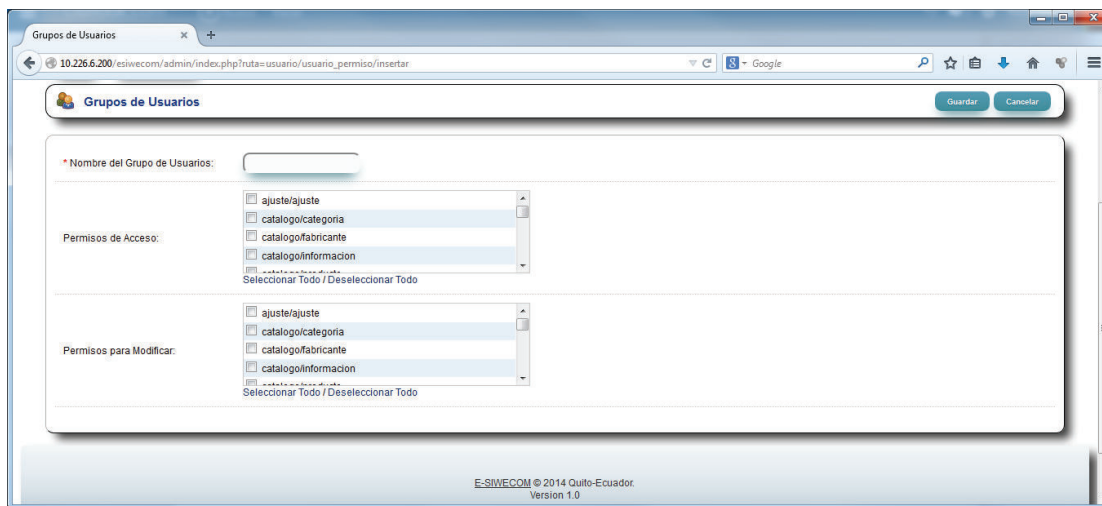


Figura 2.25: Implementación de Grupos de Usuarios (Back end)

Las figura 2.26 muestra la interfaz que responde a las funcionalidades de crear y modificar usuarios. La figura 2.27 muestra interfaz de login resultante de la creación de los módulos de administración de usuarios, clientes y perfiles.

Usuario

Guardar Cancelar

* Nombre de Usuario:

* Nombre:

* Apellidos:

E-Mail:

Grupo de Usuario: Administrador

Contraseña:

Confirmar:

Estado: Deshabilitado

Figura 2.26: Crear Usuario

Administración

192.168.0.7/esivecom/admin/index.php?ruta=general/login

ESIVECOM ADMINISTRACION

Por favor, introduzca sus datos de conexión.

Nombre de Usuario:

Contraseña:

Conectar

ESIVECOM © 2014 Quito-Ecuador.
Version 1.0

Figura 2.27: Interfaz de Login

2.4.3.4 Sprint 4 o Cuarta Iteración

A continuación se muestran los resultados de la planificación del cuarto sprint.

Sprint Backlog De La Cuarta Iteración (Sprint 4).

La Tabla 2.36 muestra las historias de usuario que forman parte del Sprint 4, sus correspondientes tareas, la duración de las mismas (en horas) y el responsable de cumplirla.

La suma del tiempo de este sprint da un total de 87 horas. Las tareas son 10 (dado que las pruebas de funcionalidad se agrupan en una sola).

Se cuenta con dos desarrolladores trabajando 4 horas por día (es decir 8 horas efectivas de trabajo). Dividimos el total de horas necesarias para las horas efectivas de trabajo ($87/8$) y obtenemos como resultado que este sprint debe tener una duración de 11 días, sin embargo se le añadirá dos días más a fin de establecer un colchón a favor del desarrollo.

HISTORIA DE USUARIO	TAREAS	TIEMPO (horas)	TAREA TOMADA POR:
Ingresar Categorías	Generar la vista	12	Fernanda
	Codificar el modelo	8	Renán
	Codificar el controlador	8	Renán
	Probar la funcionalidad	1	María Elena
Modificar Categorías	Generar la vista	12	Fernanda
	Codificar el modelo	8	Renán
	Codificar el controlador	8	Renán
	Probar la funcionalidad	1	María Elena
Eliminar Categorías	Generar la vista	12	Fernanda
	Codificar el modelo	8	Renán
	Codificar el controlador	8	Renán
	Probar la funcionalidad	1	María Elena
	Total de horas a invertir	87	
	Total de tareas	10	

Tabla 2.36: SPRINT BACKLOG 4

Burn down Chart Del Sprint 4

Al Sprint se le ha asignado una duración de 13 días, se considera un equipo dedicando al desarrollo de 2 miembros con un trabajo ideal de 4 horas al día.

En este Sprint se contabilizan 13 tareas. La figura 2.28 muestra la evolución de las tareas en el tiempo asignado.

Como se observa en la figura 2.28 la pendiente de trabajo real se muestra constante pero por sobre lo estimado. Al cierre del Sprint 4 quedan pendientes 3 tareas que deberán ser culminadas como primer paso en el Sprint 5.

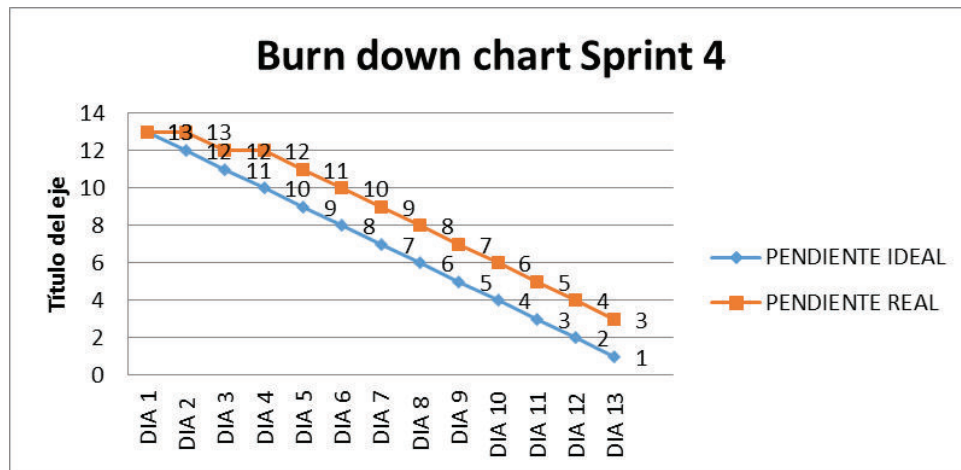


Figura 2.28: Burn down chart Sprint 4

Resultado del Sprint 4 o Cuarta iteración

Las Categorías de los productos pasan a constituir parte del menú que se decidió debía llamarse Catálogos. La figura 2.29 muestra la funcionalidad Ingresar categorías, la posibilidad de modificar o eliminar categorías se refleja en la figura 2.30.

Figura 2.29: Ingresar Categorías

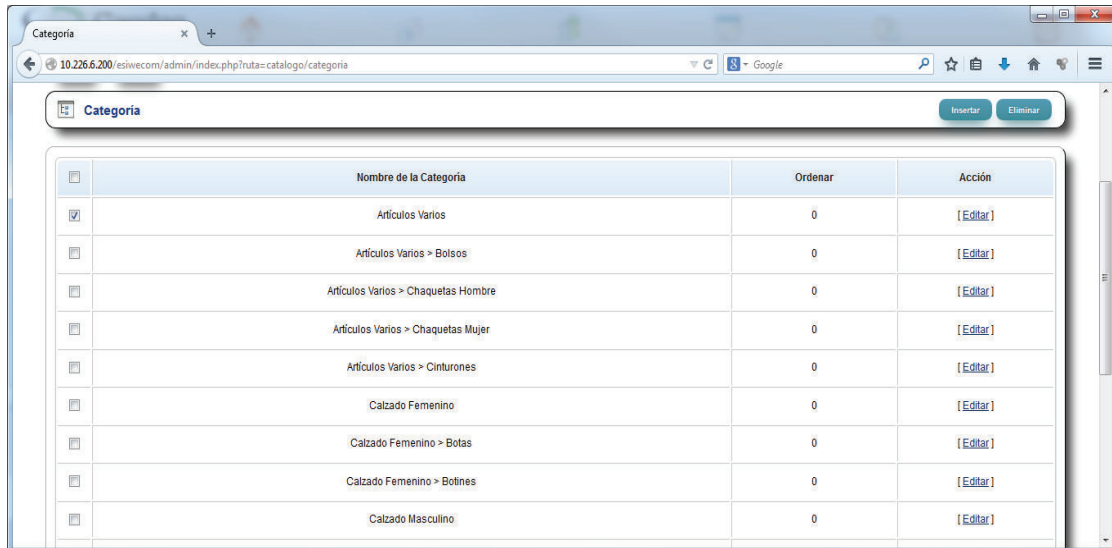


Figura 2.30: Opciones eliminar y editar categorías

2.4.3.5 Sprint 5 o Quinta Iteración

A continuación se muestran los resultados de la planificación del cuarto sprint.

Sprint Backlog De La Quinta Iteración (Sprint 5).

La Tabla 2.37 muestra las historias de usuario que forman parte del Sprint 5, sus correspondientes tareas, la duración de las mismas (en horas) y el responsable de cumplirla.

La suma del tiempo de este sprint da un total de 78 horas. Las tareas son 13 (dado que las pruebas de funcionalidad se agrupan en una sola). La disminución del tiempo asignado a las tareas se debe a la reutilización de código generado en el sprint anterior, lo cual agiliza el proceso.

Se cuenta con dos desarrolladores trabajando 4 horas por día (es decir 8 horas efectivas de trabajo). Dividimos el total de horas necesarias para las horas efectivas de trabajo (80/8) y obtenemos como resultado que este sprint debe tener una duración de 10 días, sin embargo se le añadirá dos días más a fin de establecer un colchón a favor del desarrollo.

HISTORIA DE USUARIO	TAREAS	TIEMPO (horas)	TAREA TOMADA POR:
Eliminar Categorías	Generar la vista	2	Fernanda
	Codificar el modelo	1	Renán
	Codificar el controlador	1	Renán
	Probar la funcionalidad	1	María Elena
Ingresar Productos	Generar la vista	6	Fernanda
	Codificar el modelo	3	Renán
	Codificar el controlador	3	Renán
	Probar la funcionalidad	0,5	María Elena
Modificar Productos	Generar la vista	6	Fernanda
	Codificar el modelo	3	Renán
	Codificar el controlador	3	Renán
	Probar la funcionalidad	0,5	María Elena
Eliminar Productos	Generar la vista	6	Fernanda
	Codificar el modelo	3	Renán
	Codificar el controlador	3	Renán
	Probar la funcionalidad	0,5	María Elena
Ingresar Fabricantes	Generar la vista	6	Fernanda
	Codificar el modelo	3	Renán
	Codificar el controlador	3	Renán
	Probar la funcionalidad	0,5	María Elena
Modificar Fabricantes	Generar la vista	6	Fernanda
	Codificar el modelo	3	Renán
	Codificar el controlador	3	Renán
	Probar la funcionalidad	0,5	María Elena
Eliminar Fabricantes	Generar la vista	6	Fernanda
	Codificar el modelo	3	Renán
	Codificar el controlador	3	Renán
	Probar la funcionalidad	0,5	María Elena
	Total de horas a invertir	80	
	Total de tareas	22	

Tabla 2.37: SPRINT BACKLOG 5

Burn down Chart Del Sprint 5

Al Sprint se le ha asignado una duración de 12 días, se considera un equipo dedicando al desarrollo de 2 miembros con un trabajo ideal de 4 horas al día.

En este Sprint se contabilizan 22 tareas. La figura 2.31 muestra la evolución de las tareas en el tiempo asignado.

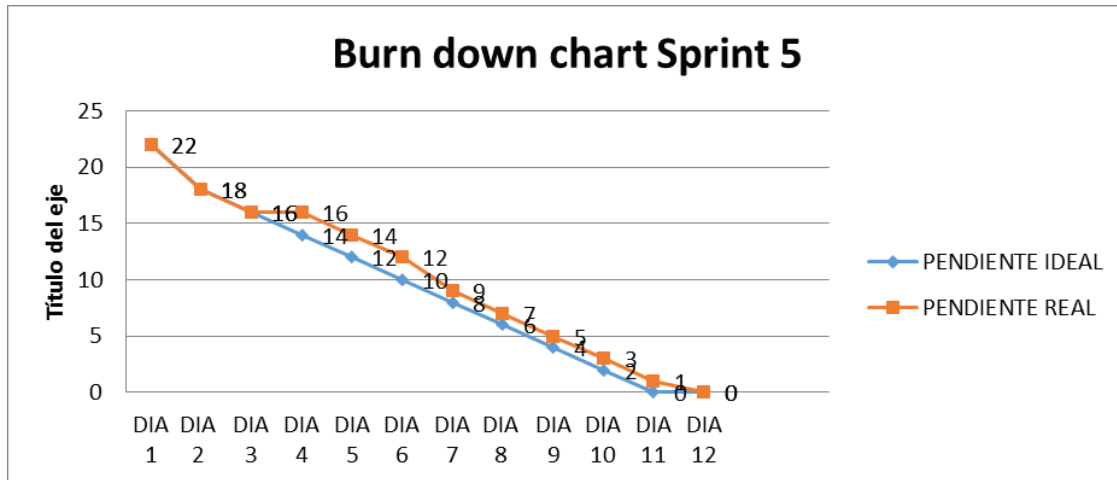


Figura 2.31: Burn down chart Sprint 5

Como se observa en la figura 2.31 la pendiente de trabajo real se muestra constante y bastante ajustada a la pendiente ideal, únicamente con un ligero retraso en la terminación de las tareas pendientes entre el día 4 y el 7, pero cuyo proceso fue posteriormente agilizado y ajustado a lo ideal. Finalmente se cerró el Sprint con las tareas cumplidas a cabalidad, aunque haciendo uso del colchón de dos días que se incluyó en el calendario.

Resultado del Sprint 5 o Quinta iteración

El resultado de la ejecución de la quinta iteración se observa en la figura 2.31. Tanto los Productos como los Fabricantes forman parte del menú Catálogos. Al momento la interfaz Productos y la interfaz Fabricantes permiten observar el contenido ya existente, editarlo o eliminarlo, dando por concluido el sprint 5 satisfactoriamente.

De este Sprint extraemos la lección que para módulos que conllevan una especie de patrón en el código que lo conforma, el tiempo puede reducirse en gran manera por la reutilización de código. Esta tarea se ve facilitada por la implementación del modelo MVC, que permite mantener un código ordenado y fácilmente escalable.

Las figura 2.32 y 2.33 muestran las interfaces para ingresar producto y modificar producto respectivamente. La figura 2.354 muestra el módulo de administración de productos.

Productos

Guardar Cancelar

General Datos Opción Descuento Oferta Imagen

* Nombre del Producto:

Meta Tag Descripción:

Descripción:

Fuente HTML

Fuente

Figura 2.32: Ingresar Productos

Productos

Guardar Cancelar

General Datos Opción Descuento Oferta Imagen

* Modelo:

SKU:

Ubicación:

Palabras clave SEO:

Imagen:

Fabricante:

Regulere Envío: SI No

Figura 2.33: Modificar Productos

Productos

Incluir Copiar Eliminar

<input type="checkbox"/>	Imagen	Nombre del Producto	Modelo	Cantidad	Estado	Acción
<input type="checkbox"/>		Artículo 1	Artículo 1	1	Habilitado	[Editar]
<input type="checkbox"/>		Bolso M1	Bolso M1	11	Habilitado	[Editar]
<input type="checkbox"/>		Bolso M2	Bolso M2	8	Habilitado	[Editar]
<input type="checkbox"/>		Bota 1	Bota 1 M1	10	Habilitado	[Editar]
<input type="checkbox"/>		Bota 2	Bota 2 M2	3	Habilitado	[Editar]
<input type="checkbox"/>		Bota 2	Bota 2 M2	2	Habilitado	[Editar]

Figura 2.34: Interfaz Administración de Productos

2.4.3.6 Sprint 6 o Sexta Iteración

A continuación se muestran los resultados de la planificación del cuarto sprint.

Sprint Backlog De La Sexta Iteración (Sprint 6).

La Tabla 2.38 muestra las historias de usuario que forman parte del Sprint 6, sus correspondientes tareas, la duración de las mismas (en horas) y el responsable de cumplirla.

HISTORIA DE USUARIO	TAREAS	TIEMPO (horas)	TAREA TOMADA POR:
Gestión Pedidos, Compras, Pagos	Generar la vista de pedidos	12	Fernanda
	Generar la vista de la gestión de pagos	12	Renán
	Codificar el controlador para pedidos	10	Renán
	Codificar el controlador para pagos	10	María Elena
	Generar el modelo para pedidos	8	Fernanda
	Codificar el modelo para pagos	8	Renán
	Probar la funcionalidad	1	María Elena
Manejo de Stock	Generar la vista	10	Fernanda
	Codificar el modelo	8	Renán
	Codificar el controlador	8	Renán
	Probar la funcionalidad	1	María Elena
Generar Información de Ventas	Generar la vista	8	Fernanda
	Codificar el modelo	10	Renán
	Codificar el controlador	10	Renán
	Probar la funcionalidad	1	María Elena
	Total de horas a invertir	119	
	Total de tareas	13	

Tabla 2.38: SPRINT BACKLOG 6

La suma del tiempo de este sprint da un total de 121 horas. Las tareas son 13 (dado que las pruebas de funcionalidad se agrupan en una sola). La disminución del tiempo asignado a las tareas se debe a la reutilización de código generado en el sprint anterior, lo cual agiliza el proceso.

Se cuenta con dos desarrolladores trabajando 4 horas por día (es decir 8 horas efectivas de trabajo). Dividimos el total de horas necesarias para las horas efectivas de trabajo ($121/8$) y obtenemos como resultado que este sprint debe tener una duración de 15 días. Por tratarse de un Sprint con métodos totalmente nuevos, no similares a los anteriores y de una complejidad mayor, se ha visto la necesidad de aumentar un colchón de tiempo mayor a los dos días que se aumentaron en los sprints anteriores, la decisión es aumentar 5 días a los ya programados.

Burn down Chart Del Sprint 6

Al Sprint se le ha asignado una duración de 20 días, se considera un equipo dedicando al desarrollo de 2 miembros con un trabajo ideal de 4 horas al día.

En este Sprint se contabilizan 13 tareas. La figura 2.35 muestra la evolución de las tareas en el tiempo asignado.

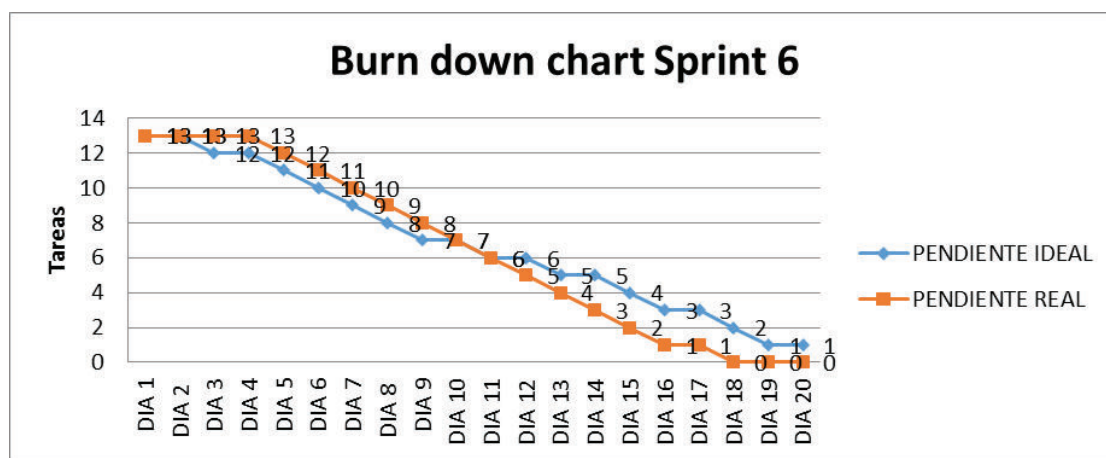


Figura 2.35: Burn down chart Sprint 6

Como se observa en la figura 2.35 la pendiente de trabajo real se ubica por sobre la pendiente de trabajo real hasta aproximadamente la mitad del sprint planteado. Desde el día 11 en adelante la pendiente de trabajo empezó a favorecer el avance del desarrollo, como se observa en la figura la pendiente real a partir de ese punto está por debajo de la pendiente ideal. Las tareas planificadas lograron culminarse en el día 18 de los 20 planteados, es decir, no se usaron 2 días de los 5 que se asignaron como colchón en favor del desarrollo.

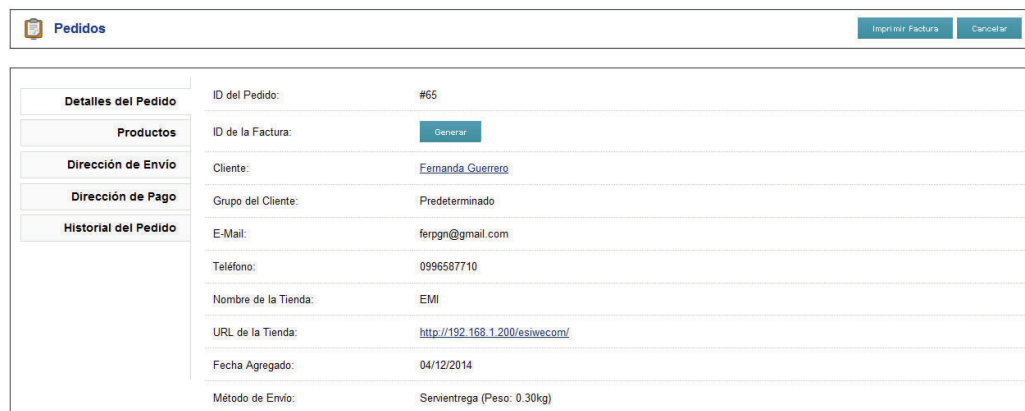
Resultado del Sprint 6 o Sexta iteración

La gestión de pedidos compras y pagos fue agrupada en una misma opción de menú llamada Pedidos. Por cada pedido realizado se genera un número identificador, y se registra el nombre del cliente, su correo electrónico, la fecha en la que se generó el pedido y el valor total, esto puede observarse en la figura 2.36. La figura 2.37 muestra que al presionar sobre la opción editar que acompaña a cada pedido, se puede observar un resumen del pedido generado que nos conduce al posterior despliegue de la factura, la factura se detalla en la figura 2.38.



ID del Pedido	Nombre del Cliente	Estado	Fecha Agregado	Total	Acción
65	Fernanda Guerrero	Pendiente	04/12/2014	\$42.56	[Editar]
64	Nataly Cansino	Procesando	03/12/2014	\$78.40	[Editar]
63	Renan Mauricio Guerrero	Procesando	03/12/2014	\$52.64	[Editar]
62	Renan Mauricio Guerrero	Pendiente	03/12/2014	\$77.28	[Editar]
61	Renan Mauricio Guerrero	Procesando	03/12/2014	\$52.64	[Editar]
60	Renan Mauricio Guerrero	Completo	03/12/2014	\$215.04	[Editar]

Figura 2.36: Menú Pedidos generado en la sexta iteración (Sprint 6)



Detalles del Pedido	ID del Pedido:	#65
Productos	ID de la Factura:	Generar
Dirección de Envío	Cliente:	Fernanda Guerrero
Dirección de Pago	Grupo del Cliente:	Predeterminado
Historial del Pedido	E-Mail:	ferpgr@gmail.com
	Teléfono:	0996587710
	Nombre de la Tienda:	EMI
	URL de la Tienda:	http://192.168.1.200/esiwecom/
	Fecha Agregado:	04/12/2014
	Método de Envío:	Servientrega (Peso: 0.30kg)

Figura 2.37: Opción Editar Pedido

EMI Quito, Ecuador Teléfono 123456789 admin@guerrero.com http://192.168.1.200/esiwecom		Fecha de la Factura: 04/12/2014 ID del Pedido: 65		
Para		Envíe a (si es diferente la dirección)		
Fernanda Guerrero Mia El Dorado Quito Pichincha Ecuador		Fernanda Guerrero Mia El Dorado Quito Pichincha Ecuador		
Producto	Modelo	Cantidad	Precio Unitario	Total
Bolso M1	Bolso M1	1	\$23.00	\$23.00
Cinturón M3	Cinturón M3	1	\$13.00	\$13.00
SubTotal:				\$36.00
Servientrega (Peso: 0.30kg):				\$2.00
12% IVA:				\$4.56
Total:				\$42.56

Figura 2.38: Generación de la Factura asociada a un pedido.

La generación de información resumen para la propietaria del negocio es entregada en dos menús, en el menú Reportes y en la página de inicio del módulo de administración a la que se nombró Dashboard

La figura 2.39 muestra parte de lo que se puede encontrar en el menú Reportes y la figura 2.40 muestra la página Dashboard.

Dashboard	Catalogo	Pedidos	Perfiles	Ajustes del Sistema	Reportes	Tienda Front-End	Desconectar
-----------	----------	---------	----------	---------------------	----------	------------------	-------------

Reporte de Productos Vistos Restablecer			
Nombre del Producto	Modelo	Visto	Porcentaje
Bota 2	Bota 2 M2	29	22.14%
Mau Zapato	Mau 1x2	19	14.5%
Zapato sin Cordón 2	Zapato sin Cordón 2 M2	16	12.21%
Zapato Cordón 1	Zapato Cordón 1 M1	12	9.16%
Bota 4	Bota 4 M2	11	8.4%
Zapato sin Cordón 1	Zapato sin Cordón 1 M1	10	7.63%
Botines 2	Botines 2 M2	7	5.34%
Zapato Cordón 2	Zapato Cordón 2 M2	5	3.82%

Figura 2.39: Reporte de productos Vistos ordenados descendientemente por porcentaje

Dashboard
Catalogo
Pedidos
Perfiles
Ajustes del Sistema
Reportes

Tienda Front-End
Desconectar

Dashboard

Información General

Ventas Totales:	\$3,901.90
Ventas Totales en este año:	\$3,901.90
Pedidos Totales:	31
No. de Clientes:	12
No. Productos:	25

Últimos 10 Pedidos

ID del Pedido	Nombre del Cliente	Estado	Fecha Agregado	Total	Acción
65	Fernanda Guerrero	Pendiente	04/12/2014	\$42.56	[Editar]
64	Nataly Cansino	Procesando	03/12/2014	\$78.40	[Editar]
63	Renan Mauricio Guerrero	Procesando	03/12/2014	\$52.64	[Editar]
62	Renan Mauricio Guerrero	Pendiente	03/12/2014	\$77.28	[Editar]
61	Renan Mauricio Guerrero	Procesando	03/12/2014	\$52.64	[Editar]

Figura 2.40: Dashboard

2.4.3.7 Sprint 7 o Séptima Iteración

A continuación se muestran los resultados de la planificación del cuarto sprint.

Sprint Backlog De La Séptima Iteración (Sprint 7).

La Tabla 2.39 muestra las historias de usuario que forman parte del Sprint 6, sus correspondientes tareas, la duración de las mismas (en horas) y el responsable de cumplirla.

suma del tiempo de este sprint da un total de 51 horas. Las tareas son 10 (dado que las pruebas de funcionalidad se agrupan en una sola). La disminución del tiempo asignado a las tareas se debe a la reutilización de código generado en el sprint anterior, lo cual agiliza el proceso.

Se cuenta con dos desarrolladores trabajando 4 horas por día (es decir 8 horas efectivas de trabajo). Dividimos el total de horas necesarias para las horas efectivas de trabajo ($51/8$) y obtenemos como resultado que este sprint debe tener una duración de 7 días. El colchón de tiempo que añadiremos en este Sprint será de 2 días.

HISTORIA DE	TAREAS	TIEMPO	TAREA TOMADA
-------------	--------	--------	--------------

USUARIO		(horas)	POR:
Ingresar Información Empresa	Generar la vista	8	Fernanda
	Codificar el modelo	4	Renán
	Codificar el controlador	4	Renán
	Probar la funcionalidad	1	María Elena
Modificar Información Empresa	Generar la vista	8	Fernanda
	Codificar el modelo	4	Renán
	Codificar el controlador	4	Renán
	Probar la funcionalidad	1	María Elena
Eliminar Información Empresa	Generar la vista	8	Fernanda
	Codificar el modelo	4	Renán
	Codificar el controlador	4	Renán
	Probar la funcionalidad	1	María Elena
	Total de horas a invertir	51	
	Total de tareas	10	

Tabla 2.39: SPRINT BACKLOG 7

Burn down Chart Del Sprint 7

Al Sprint se le ha asignado una duración de 9 días, se considera un equipo dedicando al desarrollo de 2 miembros con un trabajo ideal de 4 horas al día.

En este Sprint se contabilizan 10 tareas. La figura 20 muestra la evolución de las tareas en el tiempo asignado.

Como se observa en la figura 2.41 la pendiente de trabajo real muestra un avance en las tareas mejor al esperado. El trabajo fue eficiente al punto que el día 8 ya no habían tareas pendientes, en otras palabras, el colchón de tiempo asignado no fue utilizado.

Resultado del Sprint 7 o Séptima iteración

La información de la empresa se la incluyó en el módulo Catálogos ya existente. La figura 2.42 muestra la interfaz para Ingresar Información Empresa

De este modo se concluyen las 27 tareas planificadas al inicio del proyecto.

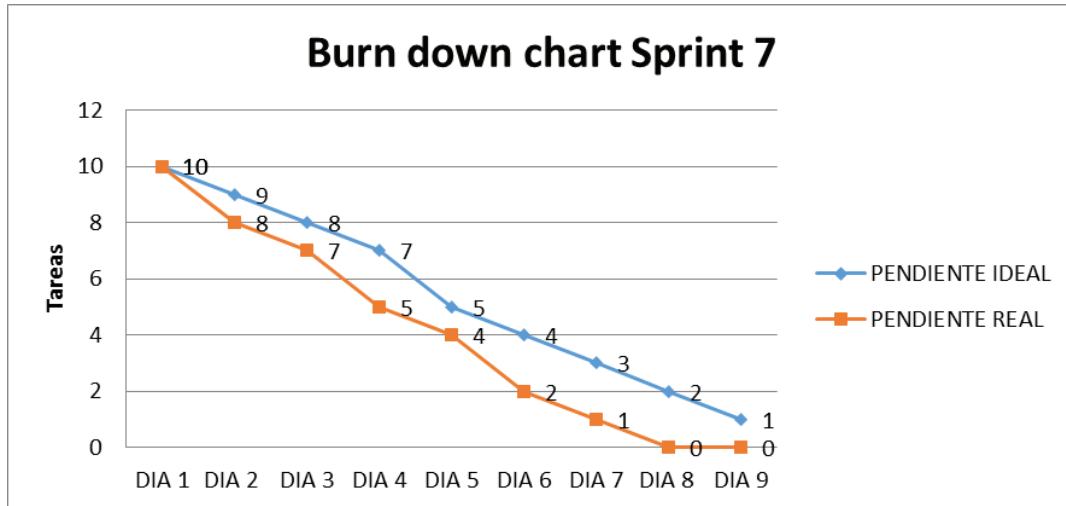


Figura 2.41: Burn down chart Sprint 7

Información [Guardar] [Cancelar]

* Título de la Información:

* Descripción:

La TIENDA VIRTUAL de EMI - EMI S.A. pone a disposición de sus clientes la posibilidad de comprar sus productos a través de Internet. Los términos y condiciones de las compras que realice el cliente por este medio serán los que constan a continuación y que serán aceptados por el cliente, como condición esencial para acceder a los productos ofrecidos por EMI - EMI S.A. por medio de este Sitio.

La posibilidad de realizar compras a través de esta página, está disponible de manera exclusiva para los clientes que cuenten con capacidad jurídica, según lo dispuesto por la normatividad colombiana vigente. En consecuencia, EMI - EMI S.A. no se encuentra obligada a emitir aceptación de una oferta emanada de un cliente que no posea la capacidad necesaria para contratar.

Independencia de disposiciones.

En caso de que una o más de las disposiciones contenidas en este documento sean consideradas nulas, ilegales o ineficaces en cualquier aspecto, la validez, legalidad y exigibilidad o eficacia del resto de las disposiciones del presente documento, no se verán afectadas o anuladas por dicha circunstancia.

Palabras clave SEO:

Estado de la Categoría:

Orden:

Figura 2.42: Ingresar Información Empresa

2.4.4 COMPARACIÓN DE LA PLANIFICACIÓN INICIAL DE LAS ITERACIONES/SPRINTS VS LA IMPLEMENTACIÓN REAL

La planificación inicial, siendo ideal, fue diferente a la implementada en términos reales. Los ajustes paulatinos en el tiempo asignado a las tareas hicieron necesario que se aumenten dos Sprints. Esta reasignación de tareas se la hace en el *Product Backlog Refinement* y el product Owner la ratifica en el posterior *Sprint Planning Meeting*

En la práctica siempre se hace necesario dedicarle tiempo a la planificación de recursos, determinando el tiempo y esfuerzo de los involucrados en el desarrollo. Pese a lo sesudo que pueda ser dicho análisis el tiempo tiende a extenderse especialmente en los primeros Sprints, pues es ahí cuando se arma el esqueleto funcional del sistema y ello demanda tiempo muchas veces subestimado.

Una vez finalizado el proyecto, todas las experiencias adquiridas a nivel de desarrollo, se analizan con todo el equipo en la reunión denominada *Sprint Retrospective*.

Las metodologías de desarrollo ágil se caracterizan por su flexibilidad y pronta respuesta al cambio o a requerimientos inesperados, y es esta misma flexibilidad la que implica alteraciones, a veces mínimas, en la planificación de las tareas y el orden en las que se ejecutan. Es por esto que la documentación no es excesiva, sino puntual.

La tabla 2.40 establece la comparación entre la planificación ideal vs la real o resultante. Inicialmente existió una planificación de 5 Sprints, pero ya en la práctica fueron necesarios 7 Sprints para conseguir terminar el sistema.

2.4.5 EJEMPLOS DEL CÓDIGO FUENTE EN BASE AL MODELO MVC

2.4.5.1 Código de las Clases Del Modelo

En esta capa se crearán las clases encargadas de manejar la interacción con la base de datos y responder a la lógica del negocio.

A continuación se de un ejemplo de la codificación del Modelo, para ello se ha extraído de la clase `ModeloUsuarioUsuario` la función `agregarUsuario`.


```

<?php
class ModeloUsuarioUsuario extends Modelo {
    public function agregarUsuario($datos) {
        $this->basedatos->consultar("INSERT INTO " . DB_PREFIJO . "usuario SET
username = " . $this->basedatos->escapar($datos['username']) . ", password = " . $this-
>basedatos->escapar(md5($datos['password'])) . ", nombres = " . $this->basedatos-
>escapar($datos['nombres']) . ", apellidos = " . $this->basedatos->escapar($datos['apellidos']) . ",
email = " . $this->basedatos->escapar($datos['email']) . ", id_usuario_grupo = " .
(int)$datos['id_usuario_grupo'] . ", estado = " . (int)$datos['estado'] . ", fecha_agregado = NOW()");
    }
}

```

SPRINTS PLANIFICADOS VS SPRINTS IMPLEMENTADOS			
ID	HISTORIA DE USUARIO	SPRINT PLANIFICADO	SPRINT IMPLEMENTADO
24	Añadir Producto a la Compra	1	1
25	Eliminar Producto	1	1
26	Crear Cuentas de Usuario	1	1
21	Realizar la Compra	1	2
20	Modificar Cuentas de Usuario	2	2
22	Eliminar Cuentas de Usuario	2	2
1	Ingresar Nuevo Usuario	2	3
2	Modificar Usuario	2	3
3	Eliminar Usuario	2	3
4	Implementación de Perfiles	2	3
5	Ingresar Categorías	3	4
6	Modificar Categorías	3	4
7	Eliminar Categorías	3	4
8	Ingresar Productos	3	5
9	Modificar Productos	4	5
10	Eliminar Productos	4	5
11	Ingresar Empresas Proveedoras	4	5
12	Modificar Empresas Proveedoras	4	5
13	Eliminar Empresas Proveedoras	4	5
14	Gestión Pedidos, Compras, Pagos	5	6
15	Manejo de Stock	5	6
16	Generar Información de Ventas	5	6
17	Ingresar Información Empresa	5	7
18	Modificar Información de la Empresa	5	7
19	Eliminar Información de la Empresa	5	7

Tabla 2.40: SPRINTS PLANIFICADOS VS SPRINTS IMPLEMENTADOS

2.4.5.2 Código de las Clases Del Controlador

El controlador es el responsable de manejar las peticiones y proveer una respuesta.

A fin de guardar coherencia con el código mostrado en el subtema Modelo, se presenta un extracto del código del controlador para la misma clase Usuario, centrando la atención en el método insertar:

```
<?php
class ControladorUsuarioUsuario extends Controlador {
    private $_error = array();

    public function index() {
        $this->cargar->etiqueta('usuario/usuario');

        $this->documento->titulo = $this->etiqueta->get('cabecera_titulo');

        $this->cargar->modelo('usuario/usuario');

        $this->getLista();
    }

    public function insertar() {
        $this->cargar->etiqueta('usuario/usuario');

        $this->documento->titulo = $this->etiqueta->get('cabecera_titulo');

        $this->cargar->modelo('usuario/usuario');

        if (($this->solicitud->servidor['REQUEST_METHOD'] == 'POST') && $this->validarForm()) {
            $this->modelo_usuario_usuario->agregarUsuario($this->solicitud->post);

            $this->sesion->datos['exito'] = $this->etiqueta->get('texto_exito');
```

```

$url = "";

if (isset($this->solicitud->get['pagina'])) {
    $url .= '&pagina=' . $this->solicitud->get['pagina'];
}

if (isset($this->solicitud->get['clasificacion'])) {
    $url .= '&clasificacion=' . $this->solicitud->get['clasificacion'];
}

if (isset($this->solicitud->get['orden'])) {
    $url .= '&orden=' . $this->solicitud->get['orden'];
}

$this->redireccionar(HTTPS_SERVIDOR .
'index.php?ruta=usuario/usuario' . $url);
}

$this->getForm();
}

```

2.4.5.3 Código de las Clases De La Vista

Los archivos que conforman la vista son los que permiten el despliegue de los formularios que llenará el usuario con sus datos. Estos datos serán enviados gracias a las funciones del controlador.

Para implementar fue necesario la utilización de JQuery, HTML y CSS

A continuación se muestra a implementación de la vista para el caso de uso Ingresar Nuevo Usuario:

```

<?php echo $cabecera; ?><?php echo $columnaizquierda; ?><?php echo
$columnaderecha; ?>
<div id="content">
  <div class="top">
    <div class="left"></div>
    <div class="right"></div>
    <div class="center">
      <h1><?php echo $cabecera_titulo; ?></h1>
    </div>
  </div>
  <div class="middle">
    <?php if ($error_advertencia) { ?>
    <div class="warning"><?php echo $error_advertencia; ?></div>
    <?php } ?>
    <form action="<?php echo str_replace('&', '&amp;', $accion); ?>" method="post"
    enctype="multipart/form-data" id="crear">
      <p><?php echo $texto_posee_cuenta; ?></p>
      <b style="margin-bottom: 2px; display: block;"><?php echo $texto_sus_detalle;
      ?></b>
      <div style="background: #F7F7F7; border: 1px solid #DDDDDD; padding: 10px;
      margin-bottom: 10px;">
        <table>
          <tr>
            <td width="150"><span class="required">*</span> <?php echo
            $entrada_nombres; ?></td>
            <td><input type="text" name="nombres" value="<?php echo $nombres; ?>" />
            <?php if ($error_nombres) { ?>
            <span class="error"><?php echo $error_nombres; ?></span>
            <?php } ?></td>
          </tr>
          <tr>
            <td><span class="required">*</span> <?php echo $entrada_apellidos; ?></td>
            <td><input type="text" name="apellidos" value="<?php echo $apellidos; ?>" />
            <?php if ($error_apellidos) { ?>
            <span class="error"><?php echo $error_apellidos; ?></span>
            <?php } ?></td>
          </tr>
          <tr>
            <td><span class="required">*</span> <?php echo $entrada_email; ?></td>
            <td><input type="text" name="email" value="<?php echo $email; ?>" />
            <?php if ($error_email) { ?>
            <span class="error"><?php echo $error_email; ?></span>
            <?php } ?></td>
          </tr>
          <tr>
            <td><span class="required">*</span> <?php echo $entrada_telefono; ?></td>
            <td><input type="text" name="telefono" value="<?php echo $telefono; ?>" />
            <?php if ($error_telefono) { ?>
            <span class="error"><?php echo $error_telefono; ?></span>
            <?php } ?></td>
          </tr>
          <tr>
            <td><?php echo $entrada_fax; ?></td>
            <td><input type="text" name="fax" value="<?php echo $fax; ?>" /></td>
          </tr>
        </table>
      </div>
    </form>
  </div>
</div>

```

```

        </tr>
    </table>
</div>
<b style="margin-bottom: 2px; display: block;"><?php echo $texto_su_direccion;
?></b>
<div style="background: #F7F7F7; border: 1px solid #DDDDDD; padding: 10px;
margin-bottom: 10px;">
    <table>
        <tr>
            <td width="150"><?php echo $entrada_empresa; ?></td>
            <td><input type="text" name="empresa" value="<?php echo $empresa; ?>"
        /></td>
        </tr>
        <tr>
            <td><span class="required">*</span> <?php echo $entrada_direccion_1; ?></td>
            <td><input type="text" name="direccion_1" value="<?php echo $direccion_1; ?>"
        />
            <?php if ($error_direccion_1) { ?>
            <span class="error"><?php echo $error_direccion_1; ?></span>
            <?php } ?></td>
        </tr>
        <tr>
            <td><?php echo $entrada_direccion_2; ?></td>
            <td><input type="text" name="direccion_2" value="<?php echo $direccion_2; ?>"
        /></td>
        </tr>
        <tr>
            <td><span class="required">*</span> <?php echo $entrada_ciudad; ?></td>
            <td><input type="text" name="ciudad" value="<?php echo $ciudad; ?>" />
            <?php if ($error_ciudad) { ?>
            <span class="error"><?php echo $error_ciudad; ?></span>
            <?php } ?></td>
        </tr>
        <tr>
            <td><?php echo $entrada_codigo_postal; ?></td>
            <td><input type="text" name="codigo_postal" value="<?php echo $codigo_postal;
?>" /></td>
        </tr>
        <tr>
            <td><span class="required">*</span> <?php echo $entrada_pais; ?></td>
            <td>
                <select name="id_pais" id="id_pais"
onchange="$('select[name='\id_provincia\']).load('index.php?ruta=cuenta/crear/provincia&
id_pais=' + this.value + '&id_provincia=<?php echo $id_provincia; ?>');">
                    <option value="FALSE"><?php echo $texto_seleccionar; ?></option>
                    <?php foreach ($paises as $pais) { ?>
                    <?php if ($pais['id_pais'] == $id_pais) { ?>
                    <option value="<?php echo $pais['id_pais']; ?>"
selected="seleccionado"><?php echo $pais['nombre']; ?></option>
                    <?php } else { ?>
                    <option value="<?php echo $pais['id_pais']; ?>"><?php echo $pais['nombre'];
?></option>
                    <?php } ?>
                </td>
    </table>

```

```

        <?php } ?>
    </select>
    <?php if ($error_pais) { ?>
    <span class="error"><?php echo $error_pais; ?></span>
    <?php } ?>

        </td>
    </tr>
    <tr>
    <td><span class="required">*</span> <?php echo $entrada_provincia; ?></td>
    <td><select name="id_provincia">
    </select>
    <?php if ($error_provincia) { ?>
    <span class="error"><?php echo $error_provincia; ?></span>
    <?php } ?></td>
    </tr>
</table>
</div>
<b style="margin-bottom: 2px; display: block;"><?php echo $texto_su_password;
?></b>
<div style="background: #F7F7F7; border: 1px solid #DDDDDD; padding: 10px;
margin-bottom: 10px;">
    <table>
    <tr>
    <td width="150"><span class="required">*</span> <?php echo
$entrada_password; ?></td>
    <td><input type="password" name="password" value="<?php echo $password;
?>" />
    <?php if ($error_password) { ?>
    <span class="error"><?php echo $error_password; ?></span>
    <?php } ?></td>
    </tr>
    <tr>
    <td><span class="required">*</span> <?php echo $entrada_confirmar; ?></td>
    <td><input type="password" name="confirmar" value="<?php echo $confirmar;
?>" />
    <?php if ($error_confirmar) { ?>
    <span class="error"><?php echo $error_confirmar; ?></span>
    <?php } ?></td>
    </tr>
    </table>
</div>

    <?php if ($texto_aceptar) { ?>
<div class="buttons">
    <table>
    <tr>
    <td align="right" style="padding-right: 5px;"><?php echo $texto_aceptar; ?></td>
    <td width="5" style="padding-right: 10px;"><?php if ($aceptar) { ?>
    <input type="checkbox" name="aceptar" value="1" checked="checked" />
    <?php } else { ?>
    <input type="checkbox" name="aceptar" value="1" />
    <?php } ?></td>

```

```

        <td align="right" width="5"><a onclick="$('#crear').submit();"
class="button"><span><?php echo $boton_continuar; ?></span></a></td>
    </tr>
</table>
</div>
<?php } else { ?>
<div class="buttons">
    <table>
    <tr>
        <td align="right"><a onclick="$('#crear').submit();" class="button"><span><?php
echo $boton_continuar; ?></span></a></td>
    </tr>
    </table>
</div>
<?php } ?>
</form>
</div>
<div class="bottom">
    <div class="left"></div>
    <div class="right"></div>
    <div class="center"></div>
</div>
</div>
<script type="text/javascript"><!--
$('select[name='id_provincial']").load('index.php?ruta=cuenta/crear/provincia&id_pais=<?
php echo $id_pais; ?>&id_provincia=<?php echo $id_provincia; ?>');
//--></script>
<?php echo $footer; ?>

```

2.4.5.4 Uso de código CSS

Para dar estilos a la vista se usó código CSS. A continuación un extracto del código generado, mismo que fue guardado en el archivo estilos.css y usado desde los archivos .php que forman parte de la vista en sus diferentes módulos.

```

html {
    margin: 0;
    padding: 0;
    height: 100%;
}
body {
    margin: 0;
    padding: 0;
    height: 100%;
    background: #FFFFFF;
}

```

```
body, td, th, input, select, textarea, option, optgroup {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px;
    color: #000000;
}
input[type='text'] {

background: rgba(249,251,252,1);
background: -moz-linear-gradient(left, rgba(249,251,252,1) 0%, rgba(249,251,252,0.82)
100%);
background: -webkit-gradient(left top, right top, color-stop(0%, rgba(249,251,252,1)),
color-stop(100%, rgba(249,251,252,0.82)));
background: -webkit-linear-gradient(left, rgba(249,251,252,1) 0%, rgba(249,251,252,0.82)
100%);
background: -o-linear-gradient(left, rgba(249,251,252,1) 0%, rgba(249,251,252,0.82)
100%);
background: -ms-linear-gradient(left, rgba(249,251,252,1) 0%, rgba(249,251,252,0.82)
100%);
background: linear-gradient(to right, rgba(249,251,252,1) 0%, rgba(249,251,252,0.82)
100%);
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#f9fbfc',
endColorstr='#f9fbfc', GradientType=1 );

.error {
    color: #FF0000;
    padding-top: 3px;
    display: block;
    font-size: 12px;
    font-weight: normal;
}
.scrollbox {
    border: 1px solid #CCCCCC;
    width: 350px;
    height: 100px;
    background: #FFFFFF;
    overflow-y: scroll;
```



```
}  
.htabs {  
    padding: 0px 0px 0px 10px;  
    height: 30px;  
    border-bottom: 1px solid #DDDDDD;  
    margin-bottom: 15px;  
}  
.htabs a {  
    border-top: 1px solid #DDDDDD;  
    border-left: 1px solid #DDDDDD;  
    border-right: 1px solid #DDDDDD;  
    background: #FFFFFF url('../imagen/tab.png') repeat-x;  
    padding: 6px 15px 7px 15px;  
    float: left;  
    font-family: Arial, Helvetica, sans-serif;
```

2.5 PRUEBAS

La metodología XP es bastante exigente en cuanto a la realización de pruebas, poniendo énfasis en que un sistema ampliamente probado en su etapa de desarrollo reducirá significativamente el número de posibles errores.

Durante el desarrollo del sistema, en el transcurso de la codificación de los componentes, se realizan las pruebas llamadas unitarias, mismas que son escritas por el desarrollador y automatizadas por alguna herramienta propia para el caso (en este proyecto se usa PHPUnit, como se describe en el siguiente punto de este acápite).

Al final de cada iteración se prueba la funcionalidad del componente con el Product Owner (quien en este caso es la futura administradora del sistema, la dueña del negocio). Si bien estas pruebas se realizan como parte del desarrollo, serán repetidas tras la implantación del sistema, para asegurar la consistencia del mismo en el nuevo ambiente, por ello, estas pruebas de funcionalidad llamadas Pruebas de Aceptación, formarán parte del capítulo 3.

2.5.1 PRUEBAS UNITARIAS

Las pruebas unitarias se constituyen en una labor muchas veces pensada como compleja y delicada, que se simplifica al usar herramientas creadas para este fin. En el caso de PHP la herramienta puede variar dependiendo del framework que se esté utilizando para el desarrollo, sin embargo la más confiable y conocida de las existente es PHPUnit - el detalle de su aplicación se dará dos subtemas más adelante -. PHPUnit permite un desarrollo basado en pruebas haciendo uso de un conjunto de aserciones (*assertions*) que ayudan en la tarea de verificar si la funcionalidad esperada se conseguirá o no. Sin embargo, al hablar de pruebas del código, es meritorio mencionar que existen herramientas de gran utilidad, como por ejemplo PHP Xdebug, que nos permiten testear la eficiencia y coherencia del código generado.

Se puede decir entonces que si bien el enfoque de PHPUnit es diferente al de Xdebug al momento de probar código, ambas herramientas se complementan permitiendo alcanzar de manera exitosa el objetivo que propone la metodología XP que hace referencia a conseguir un desarrollo del sistema basado en pruebas.

Se procede con la explicación de la funcionalidad de Xdebug y posteriormente se detalla el uso de PHPUnit.

2.5.1.1 Xdebug

Xdebug permite una exhaustiva depuración de código fuente. Xdebug es una extensión que se configura con el lenguaje PHP, es open source y su finalidad es optimizar el código generado a través de la detección de errores y la información relacionada a dicho error, posee un conjunto de funciones que permiten dar seguimiento a la acción generada por el código, pasos de la pila de ejecución, pasos de generados por las funciones, así como también un análisis del alcance del código, del rendimiento del mismo e incluso permite la detección de posibles cuellos de botella.

2.5.1.2 PHPUnit

PHPUnit se constituye en la herramienta más usada en lo referente a pruebas unitarias de código PHP. Al trabajar con PHPUnit en el desarrollo de pruebas unitarias se pretende aislar cada parte del programa, cada clase funcional y

demostrar que de manera independiente su resultado es el esperado y para ello se usa un conjunto de aserciones. Las aserciones más usadas se reflejan en la tabla 2.41. De este modo PHPUnit permite encontrar posibles problemas en etapas tempranas del desarrollo de software.

2.5.1.3 Consideraciones para la Codificación de las Pruebas Unitarias

Se empieza desarrollando un caso de prueba PHPUnit (o una clase de prueba) bajo las siguientes consideraciones obligatorias:

- La clase de prueba extenderá de la clase *PHPUnit_Framework_TestCase*, de lo contrario no se podrá acceder a los métodos integrados como por ejemplo *setUp()* y *tearDown()*.
- La clase de prueba debe tener un nombre similar a la que se está probando, por ejemplo, para probar la clase Vendedor, la clase de prueba se denominara VendedorTest.
- Los métodos de prueba deben comenzar con la palabra “*test*”, como *testValidacion()* y deben ser públicos.
- Los métodos de prueba no recibirán ningún parámetro.

Ejecución de una prueba PHPUnit

Para ejecutar se debe llamar al ejecutable `phpunit` y escoger las pruebas que son motivos de interés; a continuación se ejemplifica:

```
phpunit /path/de/clases/test/PostgresTest.php
```

La configuración de PHPUnit permite ejecutar varias pruebas a la vez e informar los resultados de todas ellas en un mismo informe.

2.5.1.4 Resultado de la ejecución de una prueba PHPUnit

Una vez ejecutada la prueba, PHPUnit retorna como datos el número de pruebas y aserciones.

Se muestra la salida a la prueba invocada en el punto anterior:

```

.
Time: 1 second
Tests: 1, Assertions: 1, Failures 0

```

La primera línea del resultado de las pruebas puede indicar:

- Éxito, que se represente con un punto (.)
- Error, devuelve F en lugar del punto.
- Prueba Incompleta, representada con una "I"
- Prueba Omitida, representada con una "S" (por Skipped, del inglés)

Ejemplo de una prueba del sistema ESIWECOM

La siguiente es la clase Postgres que permite la conexión a la base de datos en función del modelo utilizado. De esta clase se probará el método consultar.

```

<?php
final class Postgres {
    private $_enlace;

    public function __construct($hostname, $username, $password, $database) {
        if (!$this->_enlace = pg_connect('host=' . $hostname . ' user=' . $username . '
password=' . $password . ' dbname=' . $database)) {
            trigger_error('Error: No se pudo establecer conexion con la base de datos
utilizando ' . $username . '@' . $hostname);
        }

        //if (!$mysql_select_db($database, $this->link)) {
            //trigger_error('Error: Could not connect to database ' . $database);
        //}

        pg_query($this->_enlace, "SET CLIENT_ENCODING TO 'UTF8'");
    }

    public function consultar($sql) {

        // PostgreSQL-style quotes
        $sql = strtr($sql, "'", '"');

        $sql = preg_replace('/^\s*REPLACE\s*(INTO.*)$/i', 'INSERT $1', $sql);

        // Transform MySQL's INSERT...SET to normal INSERT...VALUES
        if(preg_match('/^\s*INSERT INTO[ a-zA-Z0-9_-]*SET/', $sql)) {
            $caracteres = preg_split('/(INSERT INTO[ a-zA-Z0-9_-]*SET)/',
            $sql, -1, PREG_SPLIT_NO_EMPTY | PREG_SPLIT_DELIM_CAPTURE);
            $insertar = preg_replace('/SET$/i', '', $caracteres[0]);
            $valores = $caracteres[1];
            preg_match_all("/(^[^=,]*)=((?:'[^']*'|[^=,])*)/s", $valores, $caracteres);
            $sql = $insertar . ' (' . implode(', ', $caracteres[1]) . ') VALUES (' .
            implode(', ', $caracteres[2]) . ')';
        }
    }
}

```

```

    $sql = preg_replace('/^ *(SELECT.*LIMIT) *(\d+),(\d+) *$/', '$1 $3 OFFSET
$2', $sql);
    $sql = preg_replace("/'0000-00-00'/" , ""-infinity" , $sql);
    //En la siguiente línea (36) y en la 67 (muestra la consulta exacta que
tiene problema),
    //muestra los errores que pudiera tener alguna consulta que se haga a la
base de datos.
    $recurso = pg_query($this->_enlace, $sql);

    if ($recurso) {
        if (is_resource($recurso)) {
            $i = 0;
            $datos = array();

            while ($resultado = pg_fetch_assoc($recurso)) {
                $datos[$i] = $resultado;

                $i++;
            }

            pg_free_result($recurso);
            //La clase stdClass() no tiene ni propiedades, ni métodos,
ni padre; es una clase vacía.
            //Podemos usar esta clase cuando necesitamos un objeto
genérico al que luego le podremos añadir propiedades.
            $consulta = new stdClass();
            //Añadimos tres atributos
            $consulta->fila = isset($datos[0]) ? $datos[0] : array();
            $consulta->filas = $datos;
            $consulta->numFilas = $i;

            unset($datos);

            return $consulta;
        } else {
            return true;
        }
    } else { //Y en la siguiente línea me muestra la consulta que da
problemas.
        trigger_error('Error: ' . pg_result_error($this->_enlace) . '<br />' .
$sql);
        exit();
    }
}

public function escapar($valor) {
    return pg_escape_string($this->_enlace, $valor);
}

public function contarAfectados() {
    return pg_affected_rows($this->_enlace);
}

```

```

    public function getUltimoid() {
        $consulta = $this->consultar("SELECT LASTVAL() AS `id`");

        return $consulta->fila['id'];
    }

    public function __destruct() {
        pg_close($this->_enlace);
    }
}
?>

```

A continuación se muestra la clase de prueba

```

<?php
require_once 'core/conectordb/postgres.php';
require_once 'PHPUnit/Framework/TestCase.php';

class PostgresTest extends PHPUnit_Framework_TestCase
{
    private $Postgres;

    protected function setUp()
    {
        parent::setUp();

        $this->Postgres = new Postgres('localhost', 'postgres', 'Naner123', 'esiwecom');
    }

    protected function tearDown()
    {
        $this->Postgres = null;
        parent::tearDown();
    }

    /**
     * Tests Postgres->consultar()
     */
    public function testConsultar()
    {
        $this->Postgres->consultar('SELECT * FROM ajuste');
    }
}

```

Resultado de la prueba: Éxito

Nombre de la Aserción	Resultado
<code>AssertTrue/AssertFalse</code>	Comprueba la entrada para verificar si es igual a true/false
<code>AssertEquals</code>	Comprueba el resultado frente a otra entrada en busca de coincidencias
<code>AssertGreaterThan</code>	Comprueba el resultado para ver si es mayor que un valor (también hay <code>LessThan</code> , <code>GreaterThanOrEqual</code> , y <code>LessThanOrEqual</code>)
<code>AssertContains</code>	Comprueba que la entrada contiene un valor específico
<code>AssertType</code>	Comprueba que una variable es de un cierto tipo
<code>AssertNull</code>	Comprueba que una variable es nula
<code>AssertFileExists</code>	Comprueba que un archivo existe
<code>AssertRegExp</code>	Comprueba la entrada con una expresión regular

Tabla 2.41: ASERCIONES USADAS CON MÁS FRECUENCIA EN LAS PRUEBAS PHPUNIT

3 CAPÍTULO 3. IMPLANTACIÓN Y ANÁLISIS DE RESULTADOS

Una vez concluida la etapa de desarrollo el siguiente paso es la implantación que se acompaña de pruebas necesarias para asegurar que el sistema responda como se espera.

La estrategia a ejecutar será: analizar en primera instancia los instrumentos de implantación, realizar las pruebas posteriores a la implantación, medir el grado de aceptación de los usuarios frente al sistema implantado y finalmente analizar los resultados.

3.1 ANÁLISIS DE INSTRUMENTOS DE IMPLANTACIÓN

3.1.1 SOFTWARE RECOMENDADO

Recordando que el sistema fue desarrollado en base al uso de herramientas libres, se definen los siguientes requerimientos de software en la tabla 3.1.

Software	Descripción
Sistema Operativo	Ubuntu 13 o superior
Motor de Base de Datos	PostgreSQL versión 8.3 o superior
Servidor Web	Apache 2.2.25 o superior
Intérprete PHP	PHP 5.3 o superior

Tabla 3.1: ESPECIFICACIÓN DEL SOFTWARE REQUERIDO

3.1.2 HARDWARE RECOMENDADO

La especificación de los requerimientos de hardware necesarios para la implantación del sistema ESIWECOM se detalla en la tabla 3.2.

Dispositivo	Descripción
Procesador	Velocidad mínimas de 2GHz
Memoria RAM	Mínimo 4 GB
Disco Duro	500 GB

Tabla 3.2: ESPECIFICACIÓN DEL HARDWARE REQUERIDO

3.1.3 ALOJAMIENTO VPS (VIRTUAL SERVER)

En términos generales, se consideró al Servidor Virtual Privado (VPS) como la mejor opción de alojamiento para el sistema ESIWECOM, porque permitirá que se levante el ambiente idóneo para el despliegue del sistema, sin tener que adaptar el sistema al servidor, sino customizándolo conforme a las necesidades probadas y aprobadas mediante las pruebas realizadas en un ambiente previo al de producción. Dichas pruebas se detallan en el siguiente subtema.

Como se mencionó en el capítulo 1, las opciones de alojamiento VPS son económicamente factibles como lo refleja la tabla 1.1.

3.2 PRUEBAS

Una vez culminada la etapa de desarrollo, en la que se ha probado la funcionalidad de los módulos uno a uno mientras han sido generados en cada iteración, se ha llegado al momento de la implantación del sistema. Previo al alojamiento del sitio web en el hosting VPS escogido (que es ya el ambiente de producción), se realizan nuevamente un conjunto de pruebas en un ambiente de preproducción, en el que simuló características similares a las de despliegue.

Las pruebas que se detalla en este apartado son:

- Pruebas de aceptación del sistema.
- Pruebas de compatibilidad del sistema en los navegadores más usados.
- Pruebas de rendimiento.

3.2.1 PRUEBAS DE ACEPTACIÓN

Las pruebas de aceptación son las que se basan en el uso del sistema una vez terminada su implementación.

El diseño de las pruebas fue realizado en base a los requisitos funcionales previamente especificados en las historias de usuario.

La realización de cada prueba de aceptación se resume en un cuadro con la estructura mostrada en la tabla 3.3.

Prueba de Aceptación:
Historia de Usuario:
Pasos de Ejecución:
Resultado esperado:
Evaluación de la prueba:

Tabla 3.3: PLANTILLA PRUEBAS DE ACEPTACION

Prueba de aceptación: nombre asignado a la prueba de aceptación.

Historia de Usuario: nombre de la Historia de usuario que será probada mediante la prueba de aceptación.

Pasos de Ejecución: pasos ejecutados para obtener un resultado.

Resultado Esperado: resultado que se desea para que una prueba sea exitosa.

Evaluación de la Prueba: calificación cualitativa, indica si la prueba fue Exitosa o Fallida.

3.2.1.1 Casos De Prueba

Como lo proponen varias fuentes de consulta sobre la metodología XP, es conveniente realizar un caso de prueba por cada Historia de Usuario. Durante cada iteración se realizó la comprobación de la funcionalidad que se alcanzaba hasta ese determinado momento. A continuación se detallan las pruebas de

usuario realizadas sobre el sistema como un conjunto de módulos funcionales ensamblados, por ello se documentan las pruebas en un orden similar al que aparecen las historias de usuario originalmente, no en el orden en que fueron desarrolladas.

No se documentan algunas historias de usuario que no revelan funcionalidad para el usuario final, sino más bien pasos intermedios entre lo que se ve en la interfaz y lo que el sistema hace en forma transparente al usuario.

En la Tabla 3.4 se muestra la prueba de aceptación realizada sobre el caso de uso Ingresar Nuevo Usuario.

Prueba de Aceptación: Ingresar Nuevo Usuario
Historia de Usuario: Ingresar Nuevo Usuario
<p>Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Acceder al módulo Perfiles opción Usuarios 3. Hacer click en el botón Insertar 4. Ingresar las credenciales solicitadas en la interfaz () 5. Presionar el botón GUARDAR
<p>Resultado esperado:</p> <p>Al llenar correctamente las credenciales se espera: Ingreso del nuevo usuario.</p> <p>Al llenar incorrectamente las credenciales se espera: Mensaje que evidencie los campos faltantes que deben ser llenados para el ingreso.</p>
Evaluación de la prueba: Exitosa en ambos casos.

Tabla 3.4: PRUEBA DE ACEPTACIÓN – INGRESAR NUEVO USUARIO

En la Tabla 3.5 se muestra la prueba de aceptación realizada sobre el caso de uso Modificar Usuario Existente.

Prueba de Aceptación: Modificar Usuario Existente
Historia de Usuario: Modificar Usuario Existente
<p>Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Acceder al módulo Perfiles submenú Usuarios 3. Una vez ubicado en el usuario de nuestro interés hacer click en el botón Editar junto a él. 4. Ingresar las credenciales solicitadas en la interfaz () 5. Presionar el botón GUARDAR
<p>Resultado esperado:</p> <p>Al llenar correctamente las credenciales se espera:</p> <p>Datos modificados.</p> <p>Al llenar incorrectamente las credenciales se espera:</p> <p>Mensaje que evidencie los campos faltantes que deben ser llenados para el ingreso.</p>
Evaluación de la prueba: Exitosa en ambos casos.

Tabla 3.5: PRUEBA DE ACEPTACIÓN – MODIFICAR USUARIO EXISTENTE

En la Tabla 3.6 se muestra la prueba de aceptación realizada sobre el caso de uso Eliminar Usuario

En la Tabla 3.7 se muestra la prueba de aceptación realizada sobre el caso de uso Ingresar Nuevas Categorías

Prueba de Aceptación: Eliminar Usuario
Historia de Usuario: Eliminar Usuario
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Acceder al módulo Perfiles opción Usuarios. 3. Una vez escogido el usuario de nuestro interés hacer click en el botón Eliminar de la parte superior.
Resultado esperado: Usuario eliminado.
Evaluación de la prueba: Exitosa.

Tabla 3.6: PRUEBA DE ACEPTACIÓN – ELIMINAR USUARIO

En la Tabla 3.7 se muestra la prueba de aceptación realizada sobre el caso de uso Ingresar Nuevas Categorías.

Prueba de Aceptación: Ingresar Nuevas Categorías
Historia de Usuario: Ingresar Nuevas Categorías
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Escoger el menú Catálogos opción Categoría 3. Presionar el botón Insertar. 4. Ingresar la información solicitadas en la interfaz () 5. Presionar el botón GUARDAR
Resultado esperado: Al llenar correctamente los datos obligatorios se espera: Ingreso de la nueva categoría. Al no llenar los datos obligatorios se espera: Mensaje que evidencie los campos faltantes que deben ser llenados para el ingreso.
Evaluación de la prueba: Exitosa en ambos casos.

Tabla 3.7: PRUEBA DE ACEPTACIÓN – INGRESAR NUEVAS CATEGORÍAS

En la Tabla 3.8 se muestra la prueba de aceptación realizada sobre el caso de uso Modificar Categorías.

Prueba de Aceptación: Modificar Categorías
Historia de Usuario: Modificar Categorías
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Escoger el menú Catálogos opción Categoría 3. Escoger la categoría que se desea modificar y dar click en la opción Editar. 4. Ingresar la información solicitadas en la interfaz () 5. Presionar el botón GUARDAR
Resultado esperado: Al llenar correctamente los datos obligatorios se espera: Modificación de la categoría. Al no llenar los datos obligatorios se espera: Mensaje que evidencie los campos faltantes que deben ser llenados para el ingreso.
Evaluación de la prueba: Exitosa en ambos casos.

Tabla 3.8: PRUEBA DE ACEPTACIÓN – MODIFICAR CATEGORÍAS

En la Tabla 3.9 se muestra la prueba de aceptación realizada sobre el caso de uso Eliminar Categorías.

Prueba de Aceptación: Eliminar Categoría
Historia de Usuario: Eliminar Categoría
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Acceder al módulo Catálogo opción Categorías. 3. Una vez escogida la categoría de nuestro interés hacer click en el botón Eliminar de la parte superior.
Resultado esperado: Categoría eliminada.
Evaluación de la prueba: Exitosa.

Tabla 3.9: PRUEBA DE ACEPTACIÓN – ELIMINAR CATEGORÍAS

En la Tabla 3.10 se muestra la prueba de aceptación realizada sobre el caso de uso Ingresar Nuevos Productos.

Prueba de Aceptación: Ingresar Nuevos Productos
Historia de Usuario: Ingresar Nuevas Categorías y Productos
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Escoger el menú Catálogos opción Producto. 3. Presionar el botón Insertar. 4. Ingresar la información solicitadas en la interfaz () 5. Presionar el botón Guardar.
Resultado esperado: Al llenar correctamente los datos obligatorios se espera: Ingreso de la nueva categoría. Al no llenar los datos obligatorios se espera: Mensaje que evidencie los campos faltantes que deben ser llenados para el ingreso.
Evaluación de la prueba: Exitosa en ambos casos.

Tabla 3.10: PRUEBA DE ACEPTACIÓN – INGRESAR NUEVOS PRODUCTOS

En la Tabla 3.11 se muestra la prueba de aceptación Modificar Productos.

Prueba de Aceptación: Modificar Productos
Historia de Usuario: Modificar Productos
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Escoger el menú Catálogos opción Productos 3. Escoger la categoría que se desea modificar y dar click en la opción Editar. 4. Ingresar la información solicitadas en la interfaz () 5. Presionar el botón GUARDAR
Resultado esperado: Al llenar correctamente los datos obligatorios se espera: Modificación de la categoría. Al no llenar los datos obligatorios se espera: Mensaje que evidencie campos faltantes a ser llenados para el ingreso.
Evaluación de la prueba: Exitosa en ambos casos.

Tabla 3.11: PRUEBA DE ACEPTACIÓN – MODIFICAR PRODUCTOS

En la Tabla 3.12 se muestra la prueba de aceptación Eliminar Productos.

Prueba de Aceptación: Eliminar Productos
Historia de Usuario: Eliminar Productos
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Acceder al módulo Catálogo opción Productos. 3. Una vez escogida la categoría de nuestro interés hacer click en el botón Eliminar de la parte superior.
Resultado esperado: Producto eliminado.
Evaluación de la prueba: Exitosa.

Tabla 3.12: PRUEBA DE ACEPTACIÓN – ELIMINAR PRODUCTOS

En la Tabla 3.13 se muestra la prueba de aceptación Ingresar Fabricantes.

Prueba de Aceptación: Ingresar Fabricantes
Historia de Usuario: Ingresar Fabricantes
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Escoger el menú Catálogos opción Fabricantes. 3. Presionar el botón Insertar. 4. Ingresar la información solicitadas en la interfaz () 5. Presionar el botón Guardar.
Resultado esperado: Al llenar correctamente los datos obligatorios se espera: Ingreso de la nueva categoría. Al no llenar los datos obligatorios se espera: Mensaje que evidencie los campos faltantes que deben ser llenados para el ingreso.
Evaluación de la prueba: Exitosa en ambos casos.

Tabla 3.13: PRUEBA DE ACEPTACIÓN – INGRESAR NUEVOS FABRICANTES

En la Tabla 3.14 se muestra la prueba de aceptación realizada sobre el caso de uso Modificar Fabricantes.

Prueba de Aceptación: Modificar Fabricantes
Historia de Usuario: Modificar Fabricantes
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Escoger el menú Catálogos opción Fabricantes 3. Escoger la categoría que se desea modificar y dar click en la opción Editar. 4. Ingresar la información solicitadas en la interfaz () 5. Presionar el botón GUARDAR
Resultado esperado: <p>Al llenar correctamente los datos obligatorios se espera:</p> <p style="padding-left: 40px;">Modificación de la categoría.</p> <p>Al no llenar los datos obligatorios se espera:</p> <p style="padding-left: 40px;">Mensaje que evidencie los campos faltantes que deben ser llenados para el ingreso.</p>
Evaluación de la prueba: Exitosa en ambos casos.

Tabla 3.14: PRUEBA DE ACEPTACIÓN – MODIFICAR FABRICANTES

En la Tabla 3.15 se muestra la prueba de aceptación Eliminar Fabricantes.

Prueba de Aceptación: Eliminar Fabricantes
Historia de Usuario: Eliminar Fabricantes
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Acceder al módulo Catálogo opción Fabricantes. 3. Una vez escogida la categoría de nuestro interés hacer click en el botón Eliminar de la parte superior.
Resultado esperado: <p>Categoría eliminada.</p>
Evaluación de la prueba: Exitosa.

Tabla 3.15: PRUEBA DE ACEPTACIÓN – ELIMINAR FABRICANTES

En la Tabla 3.16 se muestra la prueba de aceptación realizada sobre el caso de uso Información de Ventas.

Prueba de Aceptación: Información de Ventas
Historia de Usuario: Información de Ventas
<ol style="list-style-type: none"> 1. Pasos de Ejecución: 2. Ingresar al sistema como usuario Administrador. 3. Hacer click en el menú Reportes 4. Escoger la opción de su interés. <ol style="list-style-type: none"> a. Ventas <ol style="list-style-type: none"> i. Determinar el periodo de consulta ii. Seleccionar la unidad de tiempo por la que se agruparán las compras. iii. Seleccionar el estado de la venta iv. Presionar filtrar b. Productos más vistos c. Productos más comprados
<p>Resultado esperado:</p> <p>Al existir ventas:</p> <ul style="list-style-type: none"> - Para la opción ventas: Tabla que muestre las ventas realizadas según los filtros aplicados. - Para la opción productos vistos: Tabla que muestre los productos más vistos en un orden descendente con respecto al porcentaje de accesos a dichos productos. - Para la opción productos más comprados: Tabla que muestre los productos vendidos en un orden descendente con respecto a la cantidad de unidades vendidas acompañadas del valor total generado en la venta de dichos productos. <p>Al no existir ventas</p> <ul style="list-style-type: none"> - En todos los casos: Mensaje que diga: No hay datos para mostrar
<p>Evaluación de la prueba:</p> <p>Exitosa en ambos casos.</p>

Tabla 3.16: PRUEBA DE ACEPTACIÓN – INFORMACIÓN DE VENTAS

En la Tabla 3.17 se muestra la prueba de aceptación realizada sobre el caso de uso Ingresar Información de la Empresa.

Prueba de Aceptación: Ingresar Información de la Empresa
Historia de Usuario: Ingresar Información de la Empresa
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Hacer click en el menú Catálogos opción Información 3. Presionar el botón Insertar. 4. Llenar los campos correspondientes. 5. Presionar el botón GUARDAR
Resultado esperado: <p>Al llenar correctamente los datos pedidos en la interfaz se espera: Ingreso correcto de la información.</p> <p>Al llenar incorrectamente los datos pedidos en la interfaz se espera: Mensaje de que no se puede realizar el registro por datos faltantes</p>
Evaluación de la prueba: Exitosa en ambos casos.

Tabla 3.17: Prueba de aceptación – Ingresar Información de la Empresa

En la Tabla 3.18 se muestra la prueba de aceptación realizada sobre el caso de uso Modificar Información de la Empresa.

Prueba de Aceptación: Modificar Información de la Empresa
Historia de Usuario: Modificar Información de la Empresa
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Hacer click en el menú Catálogos opción Información 3. Escoger el bloque de información del cuál desea ingresar la información. 4. Llenar los campos correspondientes. 5. Presionar el botón GUARDAR
Resultado esperado: <p>Al llenar correctamente los datos pedidos en la interfaz se espera: Ingreso correcto de la información.</p> <p>Al llenar incorrectamente los datos pedidos en la interfaz se espera: Mensaje de que no se puede realizar el registro por datos faltantes</p>
Evaluación de la prueba: Exitosa en ambos casos.

Tabla 3.18: Prueba de aceptación – Modificar Información de la Empresa

En la Tabla 3.19 se muestra la prueba de aceptación realizada sobre el caso de uso Eliminar Información de la Empresa

Prueba de Aceptación: Eliminar Información de la Empresa
Historia de Usuario: Eliminar Información de la Empresa
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Hacer click en el menú Catálogos opción Información. 3. Ubicar el bloque de información que se desea eliminar. 4. Presionar el botón Eliminar.
Resultado esperado: <p>Información eliminada.</p>
Evaluación de la prueba: Exitosa.

Tabla 3.19: Prueba de aceptación – Eliminar Información de la Empresa

En la Tabla 3.20 se muestra la prueba de aceptación realizada sobre el caso de uso Crear Cuenta de Usuario.

Prueba de Aceptación: Crear Cuenta de Usuario
Historia de Usuario: Crear Cuenta de Usuario
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Escoger el menú Perfiles opción Usuarios 3. Hacer click en el botón Insertar 4. Ingresar las credenciales solicitadas en la interfaz () 5. Presionar el botón GUARDAR
Resultado esperado: <p>Al llenar correctamente las credenciales se espera:</p> <p>Ingreso del nuevo usuario.</p> <p>Al llenar incorrectamente las credenciales se espera:</p> <p>Mensaje de que no se puede realizar el registro por datos faltantes</p>
Evaluación de la prueba: Exitosa en ambos casos.

Tabla 3.20: Prueba de aceptación – Crear Cuenta de Usuario

En la Tabla 3.21 se muestra la prueba de aceptación realizada sobre el caso de uso Modificar Cuenta de Usuario.

Prueba de Aceptación: Modificar Cuenta de Usuario
Historia de Usuario: Modificar Cuenta de Usuario
<p>Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Escoger el menú Perfiles opción Usuarios 3. Ubicar el usuario a modificar y presionar Editar 4. Ingresar las credenciales solicitadas en la interfaz () 5. Presionar el botón GUARDAR
<p>Resultado esperado:</p> <p>Al llenar correctamente las credenciales se espera: Datos modificados y guardados..</p> <p>Al llenar incorrectamente las credenciales se espera: Mensaje de que no se puede realizar el registro por datos faltantes</p>
Evaluación de la prueba: Exitosa en ambos casos.

Tabla 3.21: Prueba de aceptación – Modificar Cuenta de Usuario

En la Tabla 3.22 se muestra la prueba de aceptación realizada sobre el caso de uso Eliminar Cuenta de Usuario

Prueba de Aceptación: Eliminar Cuenta de Usuario
Historia de Usuario: Eliminar Cuenta de Usuario
<p>Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario Administrador. 2. Escoger el menú Perfiles opción Usuarios 3. Ubicar el usuario a modificar y presionar Eliminar
<p>Resultado esperado:</p> <p>Cuenta de usuario eliminada.</p>
Evaluación de la prueba: Exitosa.

Tabla 3.22: PRUEBA DE ACEPTACIÓN – ELIMINAR CUENTA DE USUARIO

En la Tabla 3.23 se muestra la prueba de aceptación Buscar un producto.

Prueba de Aceptación: Buscar un producto
Historia de Usuario: Buscar un producto
<p>Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario registrado o no registrado 2. Escoger cualquiera de los siguientes métodos de búsqueda disponibles: <ol style="list-style-type: none"> i. Búsqueda por categorías ii. Búsqueda por palabras claves iii. Búsqueda avanzada
<p>Resultado esperado:</p> <p>Al existir productos que cumplan el criterio de búsqueda</p> <p>Imágenes de los productos.</p> <p>Al no existir productos que cumplan el criterio de búsqueda</p> <p>Mensaje que indique que no se encontraron productos bajo tal criterio.</p>
Evaluación de la prueba: Exitosa en ambos casos.

Tabla 3.23: PRUEBA DE ACEPTACIÓN – BUSCAR UN PRODUCTO

En la Tabla 3.24 se muestra la prueba de aceptación realizada sobre el caso de uso Añadir un producto al carro de compra.

Prueba de Aceptación: Añadir un producto al carro de compra
Historia de Usuario: Añadir un producto al carro de compra
<p>Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario registrado o no registrado 2. Escoger el producto 3. Determinar la cantidad deseada 4. Presionar añadir al carrito
<p>Resultado esperado:</p> <p>Al escoger una cantidad dentro del stock</p> <p>Producto añadido sin advertencia</p> <p>Al escoger una cantidad fuera del stock</p> <p>Producto añadido con la advertencia de que no existe dicho producto en las cantidades deseadas.</p>
Evaluación de la prueba: Exitosa en ambos casos.

Tabla 3.24: PRUEBA DE ACEPTACIÓN – AÑADIR UN PRODUCTO AL CARRO DE COMPRA

En la Tabla 3.25 se muestra la prueba de aceptación realizada sobre el caso de uso Eliminar un producto antes de Realizar una compra.

Prueba de Aceptación: Eliminar un producto antes de Realizar una compra
Historia de Usuario: Eliminar un producto antes de Realizar una compra
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario registrado o no registrado 2. Previamente haber agregado productos al carrito. 3. Acceder al menú Carrito 4. Seleccionar el producto a eliminar en la columna denominada Quitar. 5. Presionar el botón actualizar.
Resultado esperado: Producto retirado del carrito
Evaluación de la prueba: Exitosa.

Tabla 3.25: Prueba de aceptación – Eliminar un producto antes de Realizar una compra

En la Tabla 3.26 se muestra la prueba de aceptación realizada sobre el caso de uso Realizar una compra.

Prueba de Aceptación: Realizar una compra
Historia de Usuario: Realizar una compra
Pasos de Ejecución: <ol style="list-style-type: none"> 1. Ingresar al sistema como usuario registrado o no registrado 2. Previamente haber agregado productos al carrito. 3. Acceder al menú Comprar <ol style="list-style-type: none"> a. El usuario no registrado deberá crear una cuenta para proceder con los siguientes pasos. b. El usuario registrado deberá escoger método de envío y forma de pago. 4. Presionar el botón aceptar.
Resultado esperado: Mensaje de que la compra ha sido realizada con éxito y que recibirá una notificación al correo con el detalle de su compra.
Evaluación de la prueba: Exitosa.

Tabla 3.26: Prueba de aceptación – Realizar una compra

3.2.2 PRUEBAS DE COMPATIBILIDAD DEL SISTEMA CON LOS EXPLORADORES MÁS USADOS

El sistema fue accedido desde 3 diferentes exploradores, los cuales se consideran de uso más común. Estos navegadores fueron: Chrome versión 39.0, Firefox 33.1, Internet Explorer versión 8. Lo que se valoró fue la facilidad de manejo del sistema, la apariencia, el tiempo de respuesta. Se estableció una escala cuantitativa de valoración para estas características que incluye las opciones: mala, regular, buena. Los resultados obtenidos se detallan en la tabla que forma parte del subtema Análisis de resultados de este capítulo.

La figura 3.1 muestra la página principal del sistema en el navegador Chrome.

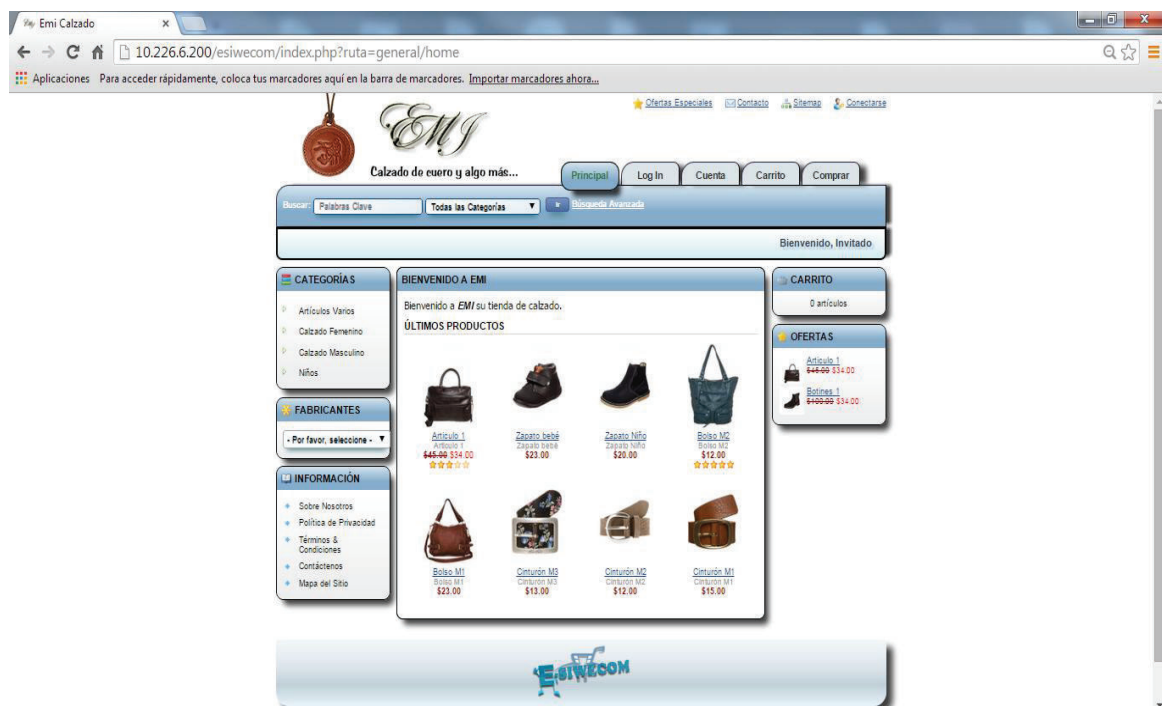


Figura 3.1: Página principal del sistema vista en Chrome 39

La figura 3.2 muestra la página principal del sistema en el navegador Firefox.

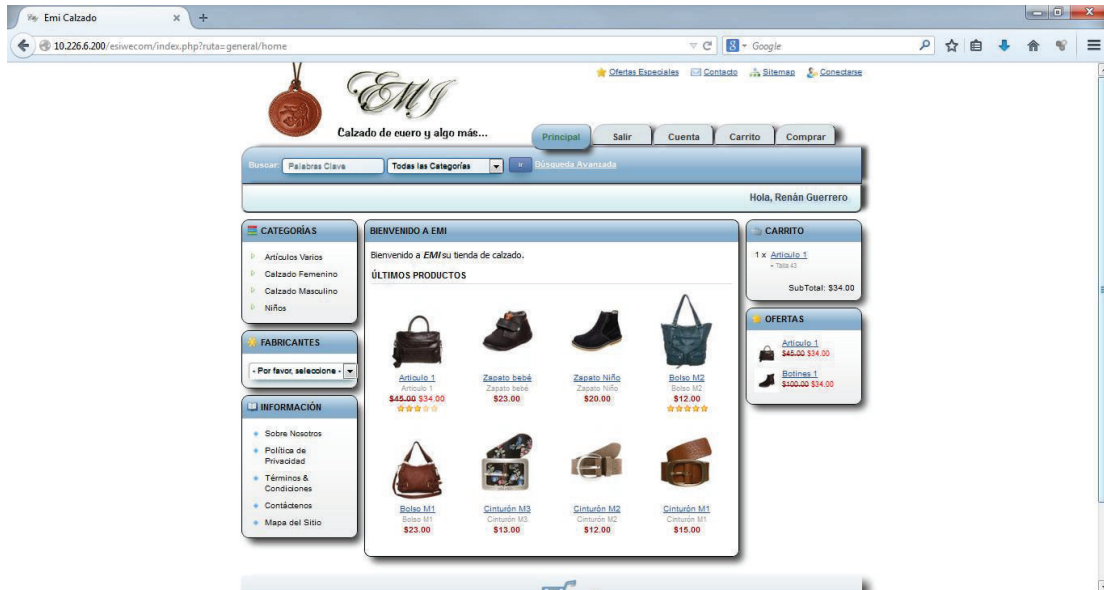


Figura 3.2: Página principal del sistema vista en Firefox 33.1

La figura 3.3 muestra la página de inicio en el navegador Internet Explorer.

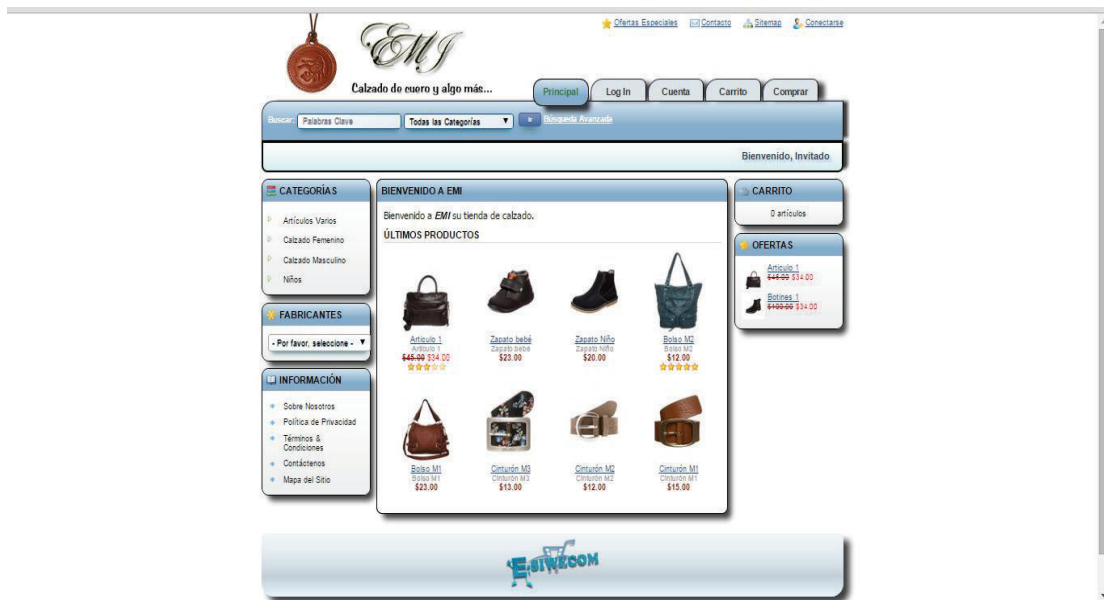


Figura 3.3: Página principal del sistema vista en Internet Explorer 8

La tabla 3.27 recoge las impresiones generadas sobre la apariencia y navegabilidad del sistema en los diferentes exploradores.

	Internet Explorer 8.0	Chrome 39	Firefox 33.1
Apariencia	Buena	Buena	Buena
Navegabilidad	Buena	Buena	Buena
Tiempo de respuesta	Buena	Buena	Buena

Tabla 3.27: EVALUACIÓN DE LAS CARACTERÍSTICAS DEL SISTEMA VISTAS EN LOS DIFERENTES NAVEGADORES

En el caso de los exploradores Chrome y Firefox se comprobó que al usar las versiones inmediatas inferiores a las que constan en la tabla 3.27, los resultados siguieron siendo favorables, no sucediendo lo mismo con el explorador Internet Explorer, en el cual la interfaz gráfica sufría una leve distorsión haciendo que su apariencia ya no sea buena.

3.2.3 PRUEBAS DE RENDIMIENTO

Las pruebas de rendimiento se realizan con el afán de determinar si el usuario final estará a gusto con la velocidad de la aplicación dibujando un escenario similar al que tendrán en condiciones normales.

En el campo del comercio electrónico el rendimiento del sistema es esencial pues los clientes no vuelven a un sitio web con bajo rendimiento.

Para probar el rendimiento del sistema implantado se usó una herramienta libre de uso online, la cual puede accederse desde la siguiente url: www.loadimpact.com.

Para medir el tiempo de respuesta el primer paso es realizar una petición en la página principal. Los resultados se grafican en la figura 3.4. La simbología es la

siguiente: las sesiones activas se representan con la línea verde y el tiempo de respuesta con la línea azul.

Se puede observar que la prueba de rendimiento es favorable pues la capacidad del servidor permite un tiempo de respuesta ágil para un número alto de conexiones activas.

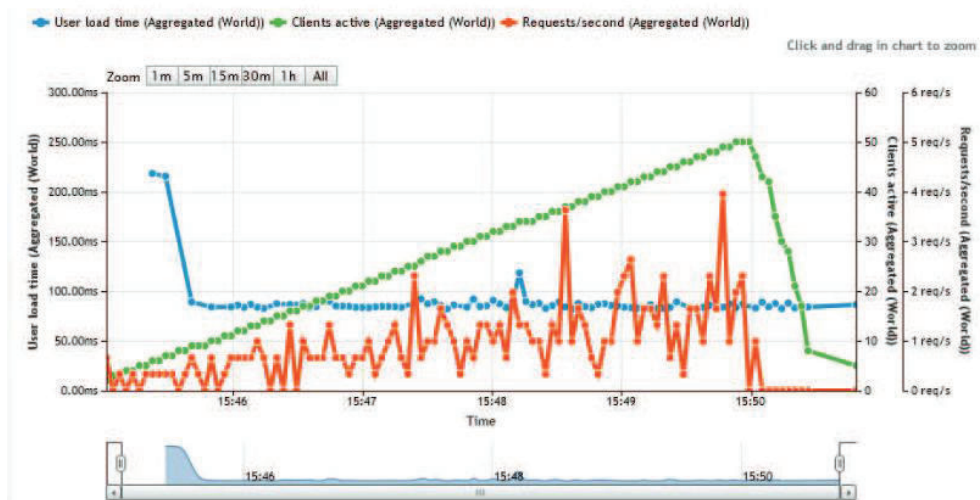


Figura 3.4: Prueba De Rendimiento - Tiempo De Respuesta

3.3 ACEPTACIÓN

Si bien las pruebas de aceptación, que resultaron exitosas, permitirían concluir que el cliente está satisfecho con el sistema, se ha creído conveniente obtener una impresión global del funcionamiento y la apariencia del producto, que incluya la opinión no solo de la futura administradora del sistema, sino también a un pequeño grupo de usuarios finales, que podrían constituirse en potenciales clientes de la tienda de calzado y artículos de cuero EMI.

Se ha diseñado una encuesta corta, que abarca aspectos puntuales y de importancia para los usuarios del sitio web. A continuación se citan las preguntas aplicadas a un conjunto de 15 usuarios finales, a los que se les contactó vía correo electrónico y tras asignarles un nombre de usuario y una contraseña para que accedan como usuarios registrados, evaluaron su experiencia de uso en un formulario diseñado a través de la herramienta Google Drive.

Las preguntas de la encuesta y sus opciones de respuesta se citan a continuación:

1. La navegación en el sitio web ESIWECOM fue fácil:

SI	NO	PARCIALMENTE
----	----	--------------
2. Le pareció agradable la apariencia del sitio web:

SI	NO	PARCIALMENTE
----	----	--------------
3. La información descriptiva de los productos le resultó completa:

SI	NO	PARCIALMENTE
----	----	--------------
4. Diría que la navegación por el sitio web fue rápida

SI	NO	PARCIALMENTE
----	----	--------------
5. En base a su experiencia usando de navegación, recomendaría este sitio web a un amigo :

SI	NO	
----	----	--

Las respuestas a esta encuesta se analizan en el apartado 3.4.

3.4 ANÁLISIS DE RESULTADOS

En este apartado se analizan los resultados de las pruebas de funcionalidad o aceptación y también los resultados de la encuesta realizada. De este modo se podrá llegar a una conclusión que permita saber si el sistema satisface las expectativas no solo del propietario de la tienda de artículos de cuero, sino también de sus clientes.

3.4.1 RESULTADOS DE LAS PRUEBAS DE ACEPTACIÓN (FUNCIONALIDAD)

La Tabla 3.28 agrupa los resultados de las pruebas realizadas por cada módulo. La conclusión a breves rasgos es que el Product Owner está satisfecho con el funcionamiento del sistema pues todas las pruebas resultaron exitosas.

De manera general, en base a un conteo, se concluye que de 26 pruebas realizadas, todas fueron casos de éxito.

	Historia de Usuario	Resultado Obtenido
1	Ingresar Nuevo Usuario	Éxito
2	Modificar Usuario	Éxito
3	Eliminar Usuario	Éxito
4	Implementación de Perfiles	Éxito
5	Ingresar Categorías	Éxito
6	Modificar Categorías	Éxito
7	Eliminar Categorías	Éxito
8	Ingresar Productos	Éxito
9	Modificar Productos	Éxito
10	Eliminar Productos	Éxito
11	Ingresar Empresas Proveedoras	Éxito
12	Modificar Empresas Proveedoras	Éxito
13	Eliminar Empresas Proveedoras	Éxito
15	Gestión Pedidos, Compras, Pagos	Éxito
16	Manejo de Stock	Éxito
17	Generar Información de Ventas	Éxito
18	Ingresar Información Empresa	Éxito
19	Modificar Información de la Empresa	Éxito
20	Eliminar Información de la Empresa	Éxito
21	Crear Cuentas de Usuario	Éxito
22	Modificar Cuentas de Usuario	Éxito
23	Eliminar Cuentas de Usuario	Éxito
24	Buscar Producto	Éxito
25	Añadir Producto a la Compra	Éxito
26	Eliminar Producto	Éxito
27	Realizar la Compra	Éxito

Tabla 3.28: RESUMEN DE LOS RESULTADOS DE LAS PRUEBAS DE ACEPTACIÓN

3.4.2 RESULTADOS DE LA ENCUESTA A LOS USUARIOS FINALES

La figura 49 muestra los resultados obtenidos respecto a la pregunta 1 y los resultados tabulados se resumen en la tabla 3.29.

Pregunta 1: La navegación en el sitio web ESIWECOM fue fácil:

RESPUESTA	FRECUENCIA
SI	8
NO	3
PARCIALMENTE	4

Tabla 3.29: Respuestas a la pregunta 1

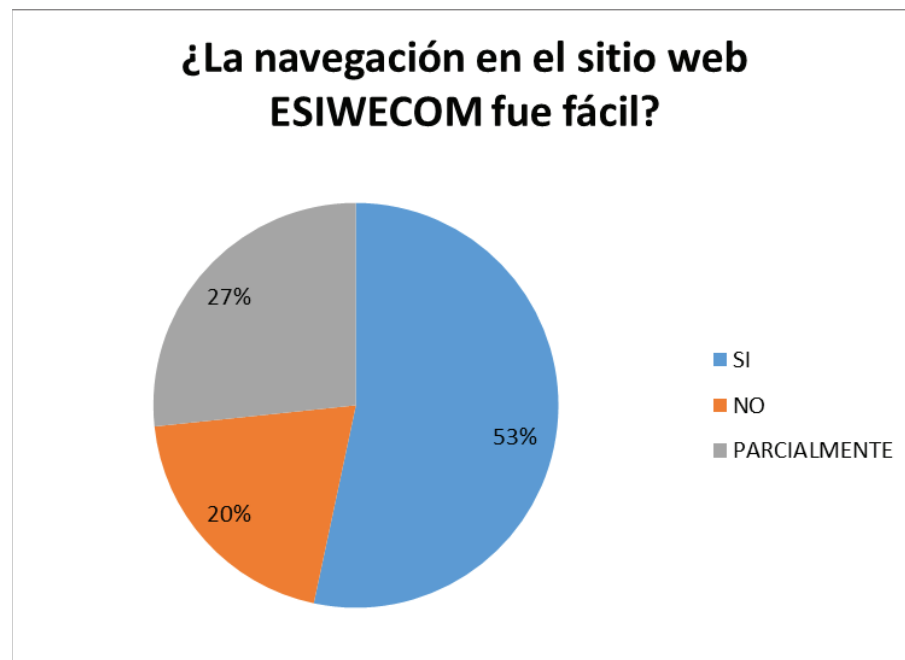


Figura 3.5: Representación gráfica de las respuestas a la pregunta 1

La figura 3.6 muestra los resultados obtenidos respecto a la pregunta 2 y los resultados tabulados se resumen en la 3.30.

Pregunta 2: ¿Le pareció agradable la apariencia del sitio web?

RESPUESTA	FRECUENCIA
SI	10
NO	2
PARCIALMENTE	3

Tabla 3.30: Respuestas a la pregunta 2.

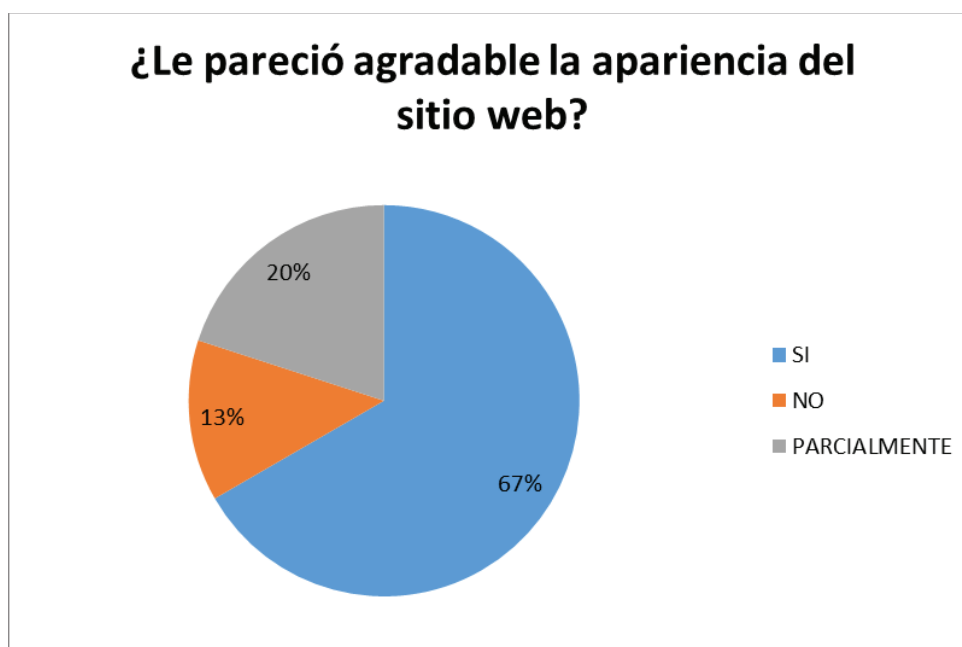


Figura 3.6: Representación gráfica de las respuestas a la pregunta 2.

La figura 3.7 muestra los resultados obtenidos respecto a la pregunta 3 y los resultados tabulados se resumen en la tabla 3.31.

Pregunta 3: ¿La información descriptiva de los productos le resultó completa?

RESPUESTA	FRECUENCIA
SI	9
NO	3
PARCIALMENTE	5

Tabla 3.31: Respuestas a la pregunta 3

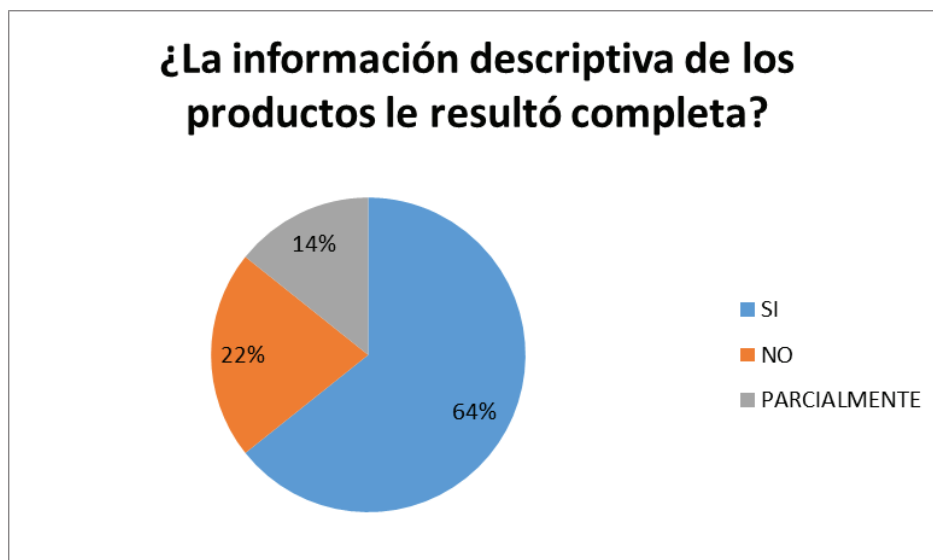


Figura 3.7: Representación gráfica de las respuestas a la pregunta 3.

La figura 3.8 muestra los resultados obtenidos respecto a la pregunta 4 y los resultados tabulados se resumen en la tabla 3.32.

Pregunta 4: ¿Diría que la navegación por el sitio web fue rápida?

RESPUESTA	FRECUENCIA
SI	11
NO	1
PARCIALMENTE	3

Tabla 3.32: Respuestas a la pregunta 4.

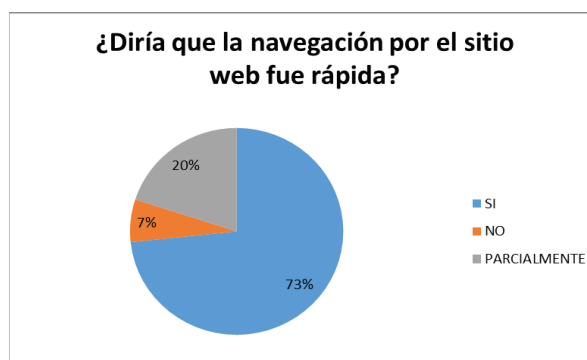


Figura 3.8: Representación gráfica de las respuestas a la pregunta 4.

La figura 3.9 muestra los resultados obtenidos respecto a la pregunta 5 y los resultados tabulados se resumen en la tabla 3.33.

Pregunta 5: En base a su experiencia de navegación, ¿recomendaría este sitio web a un amigo?:

RESPUESTA	FRECUENCIA
SI	13
NO	2

Tabla 3.33: Respuestas a la pregunta 5.



Figura 3.9: Representación gráfica de las respuestas a la pregunta 5.

De forma general se concluye que el sistema es del agrado de los usuarios finales porque resulta fácil de usar, visualmente agradable y con la información de los productos necesaria para su elección. Esta conclusión se ratifica al observar que un 87% de los encuestados sí referirían el sitio web a un amigo, lo que indica una amplia aceptación del sistema implantado.

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

En el comercio electrónico de tipo B2C (Business to Customer) intervienen la empresa que ofrece el producto y el consumidor final que requiere de él y el sistema que permite la visualización de productos ofertados y facilita la transacción entre cliente y negociante.

Los requerimientos de un sistema e-commerce, como en prácticamente todo sistema web, van encaminados a facilitar el acceso a los productos o servicios ofrecidos en base a la promoción de los mismos. La definición de requerimientos es una tarea de vital importancia, pues de ellos depende la conformidad y satisfacción de los usuarios del sistema.

Scrum es una metodología con la que los miembros del equipo de desarrollo llegan a sentirse bastante cómodos, especialmente porque deja de lado ciertos formalismos y promulga el trabajo en libertad de elección, haciendo que llevar a cabo las tareas escogidas por cada miembro sea una labor agradable y no tediosa, probablemente en esto radique el actual éxito de Scrum como metodología de desarrollo ágil.

Tanto XP como SCRUM, enfatizan la ejecución de pruebas a lo largo del ciclo de desarrollo del software, la idea central de ello es que invertir tiempo probando el sistema en etapas tempranas puede significar un ahorro de tiempo durante la vida útil del producto; esta realidad es aún más cierta al tratarse de un sistema de comercio electrónico, pues en este escenario las pruebas no solo garantizan ahorro de tiempo, sino que pueden prevenir futuras pérdidas económicas, ya que un sistema con fallas, o lento, o poco amigable, es no atractivo al usuario final, haciendo que un cliente no vuelva a usar el sistema e-commerce debido a la falta de confianza que tiene en él.

El desarrollo de software basado en pruebas –como el realizado con PHPUnit, si bien garantiza la detección pronta de errores y por ende su rápida corrección, puede tornarse en un esfuerzo no apto para equipos de desarrollo pequeños, esto debido a que es necesario el desarrollo de al menos una clase de prueba por cada clase objetivo, ya que si desea probarse tanto un caso de éxito y un caso de falla, se deben implementar una clase de cada prueba en cada caso. En equipos pequeños como el nuestro, conformado por apenas 2 personas, ésta se convierte en una labor titánica.

Las metodologías de desarrollo ágil se caracterizan por su flexibilidad y pronta respuesta al cambio o a requerimientos inesperados, y es esta misma flexibilidad la que implica alteraciones, a veces mínimas, en la planificación de las tareas y el orden en las que se ejecutan. Es por esto que la documentación no es excesiva, sino puntual. A este respecto en Scrum tanto el Sprint Backlog, el Scrum Task Board y el Burn down chart son herramientas de gran utilidad en el momento mismo en que son usadas, permitiendo el seguimiento al trabajo planteado y el conseguido.

Al hablar del negocio y la satisfacción del cliente es necesario esforzarse en dar un buen servicio, lo que implica conocer lo que el público objetivo espera del sistema, acertar en el diseño de un sistema intuitivo para que sea de fácil acceso y atraiga a la gente, así se asegura la comodidad del cliente, lo que puede traducirse en una nueva compra asegurada y publicidad generada que atraiga a más usuarios.

El desarrollo de sistemas usando el patrón MVC supone a la larga una ventaja competitiva pues son altamente escalables y permiten mantener un código más ordenado que será fácil de mantener.

4.2 RECOMENDACIONES

Si bien poner a disposición de toda la internet los productos que ofrece una tienda permite aumentar el espectro de posibles futuros clientes, factores como la

visibilidad de los productos y la asequibilidad de su precio son también elementos necesarios para el éxito de un sistema de compras on line.

En el transcurso del desarrollo de todo el proyecto es necesaria la participación activa del Producto Owner, hay que incentivar la clarificación de requerimientos, usando incluso técnicas tan sencillas como la implementación de interfaces a mano alzada, a fin de identificar lo que el requirente aspira obtener como sistema.

Al adoptar Scrum como metodología de gestión de proyectos, no se deben descuidar los Daily Meeting, aún si el equipo no está físicamente junto, pues se puede valer del teléfono, del computador, o de los teléfonos inteligentes, a fin de diariamente dar contestación a las estas tres preguntas: ¿Qué has hecho desde ayer?, ¿Qué es lo que harás hasta la reunión de mañana?, ¿Se ha suscitado algún problema que te haya impedido alcanzar tu objetivo?. Dar contestación a estas interrogantes permite proyectar lo alcanzado hasta el momento y lo que se hará, siendo un impulso constante para no declinar ni disminuir el paso, además de ser útil también para descentralizar el conocimiento.

Siendo que cada vez es mayor la gama de pruebas que se ofertan para software, se incentiva la realización de un análisis previo de las necesidades reales del sistema a implantar, conociendo su posible concurrencia, carga, y sabiendo de antemano, por la naturaleza del sistema, si las operaciones sobre el sistema (reflejadas a nivel de base de datos) serán en su mayoría lecturas, inserciones o actualizaciones, todo esto permitirá hacer la mejor elección al momento de testear y optimizar nuestros sistemas.

Es muy importante que los integrantes del equipo de desarrollo dediquen tiempo a planificar qué clases del sistema merecen testearse a nivel de pruebas unitarias con herramientas como PHPUnit, y al momento de ejecutar dichas pruebas tener presente que es conveniente no solamente establecer casos de prueba de éxito, sino también de intencional error a fin de observar cómo responde el sistema en dichos escenarios.

La planificación de recursos debe ser tomada en serio, determinando el tiempo y esfuerzo de los involucrados en el desarrollo, no inclinándose siempre por el

mejor de los escenarios sino siendo precavidos e incluyendo colchones tiempo que puedan ser un soporte en caso de tareas imprevistas en el desarrollo, pues pese a lo sesudo que pueda ser dicho análisis el tiempo tiende a extenderse especialmente en los primeros Sprints, pues es ahí cuando se arma el esqueleto funcional del sistema y ello demanda tiempo muchas veces subestimado.

Para conseguir sistemas livianos y visualmente atractivos, pueden usarse CSS3 y HTML5, que no sobrecargan al sistema y pueden hacerlo ágil en su manejo.

Al empezar el desarrollo de un sistema usando el patrón MVC el equipo de desarrollo debe poseer al menos un nivel intermedio de conocimiento de orientación a objetos, mientras más dominio sobre orientación a objetos, más productividad habrá al momento de desarrollar.

GLOSARIO

El siguiente listado corresponde a los términos usados con más frecuencias en las metodologías Scrum y XP:

Burndown Chart.- Es el gráfico que actualiza el equipo en las reuniones de seguimiento del sprint, para monitorizar el ritmo de avance, y detectar de forma temprana posibles desviaciones sobre la previsión que pudieran comprometer la entrega al final de sprint.

Historia de Usuario.- Descripción de una funcionalidad que debe incorporar un sistema de software, y cuya implementación aporta valor al cliente.

Product backlog .- Es la lista ágil de los requisitos del cliente o requisitos del sistema, estos requisitos son expresados en forma simple, son estimados y priorizados.

Product Owner.- Es el nombre dado a uno de los roles de Scrum. El propietario del producto (product owner) es quien toma las decisiones del cliente. Su responsabilidad es el valor del producto.

Prueba unitaria.- Es una forma de comprobar el correcto funcionamiento de un módulo de código para asegurar que cada uno de los módulos funcione correctamente por separado

Scrum Master: es quien lidera al equipo de trabajo dentro del proyecto, y entre sus responsabilidades está el de facilitar que las reuniones de Scrum se realicen de manera planificada.

Sprint backlog.- Es la lista de tareas que va a realizar el equipo en una iteración, para construir un incremento. Esta lista incluye el nombre de quien realizará cada tarea y una estimación del tiempo que invertirá en terminarla.

Team: es el grupo de personas que desarrollan el producto de software del

XP.- Programación Extrema (eXtreme Programming), es una metodología ágil que pone más énfasis en la adaptabilidad que en la previsibilidad

BIBLIOGRAFIA

LIBROS – REVISTAS

1. ABRAHAMSSON, Pekka, SALO, Outi, RONKAINEN , Jussi, WARSTA, Juhani. Agile Software Development Methods. VTT. 2002.
2. AHMAD Faudziah, BAHAROM Fauziah, HUSNI Moath. Agile Development Methods for Developing Web Application in Small Software Firms. Knowledge Management International Conference (KMICe). 2012.
3. ANDAM , Zorayda Ruth. E-Commerce and e-Business. 2003.
4. AWAD ,M. A Comparison between Agile and Traditional Software Development Methodologies. School of Computer Science and software Engineering, The University of Western Australia. 2005.
5. BAHIT Eugenia. Scrum y Extreme Programming para programadores. 2012.
6. BECK Kent. Extreme Programming Explained. Addison Wesley. 1999.
7. BELKINA Maria, SCHILLER Jennifer, MASUNOV Maxim, FILIPPOV Vycheslav. Scrum. 2006.
8. BOEHM , Barry W. Software Engineering Economics. IEEE. 1984.
9. BOZO PARRAGUEZ Jorge, CRAWFORD LABRIN Broderick. Métodos Ágiles como alternativa al Proceso de Desarrollo Web. Pontificia Universidad Católica de Valparaiso. Chile. 2003.
10. BRITO ACUÑA Kareny. Selección de Metodologías de Desarrollo para Aplicaciones Web. Universidad de Cienfuegos “Carlos Rafael Rodríguez” – Facultad de Informática. Cuba. 2009.
11. COCKBURN, Alistar. Agile Software Development. Highsmith Series. 2001.
12. COHN Mike. Agile Estimating and Planning. Pearson Education. 2006.
13. COHN Mike. Succeeding with Agile: Software Development Using Scrum. Addison-Wesley. 2010.
14. COHN Mike. User Stories Applied for Agile Software Development. Addison Wesley. 2004.

15. DEEMER Pete, BENEFIELD Gabrielle, LARMAN Craig, VODDE Bas. The Scrum Primer. Scrum Training Institute. 2009.
16. DESHPANDE Yogesh, MURUGESAN San, GINIGE Athula, HANSEN Steve, SCHWABE Daniel, GAEDKE Martin, WHITE Bebo. Web Engineering. Rinton Press. 2002.
17. DUTT , Saikat. Proper Product Backlog Prioritization. Cognizant. 2012.
18. E. Miriam JIMENEZ, ORANTES Sandra. Metodologías Híbridas para desarrollo de Software: Una opción factible para México. UNAM. 2012.
19. ESCALONA María José, KOCH Nora. Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo. Universidad de Sevilla, Departamento de Lenguajes y Sistemas Informáticos. 2002.
20. FARRELL, Adrienne. Selecting a Software Development Methodology based on Organizational Characteristics. Athabasca University. 2007.
21. FIGUEROA Roberth, SOLÍS Camilo, CABRERA Armando. Metodologías Tradicionales Vs. Metodologías Ágiles. Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación. 2011.
22. GAMMA, Erich, HELM, Richard. JOHNSON Ralph, VLISSIDES John. Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley. 1998.
23. GRENNING James. Planning Poker or How to avoid analysis paralysis while release planning. 2002.
24. HNEIF Malik, HOCK OW Siew. Review of Agile Methodologies in Software Development. University of Malaya, Kuala Lumpur. International Journal of Research and Reviews in Applied Sciences. 2009.
25. HOWCROFT , Debra, CARROLL , John. A Proposed Methodology for Web Development. University of Salford. 2000.
26. IMREH, Robert, RAISINGHANI, Mahesh. Impact of Agile Software Development on Quality within Information Technology Organizations. Journal of Emerging Trends in Computing and Information Sciences. 2011.
27. JACOBSON Ivar, WEI Ng Pan, and SPENCE Ian. The Essential Unified Process - A fresh start for process. 2006.
28. JALOTE Pankaj, An Integrated Approach to Software Engineering. Tercera Edición. Springer Science Business Media. 2005.

29. JIJENA LEIVA , Renato. Comercio, derecho, firma y documentos digitales o electrónicos. 1999.
30. JIMENEZ, Miriam, ORANTES , Sandra. META: a new hybrid methodology to software development created to suit the current needs in Mexico. International Institute of Informatics and Systemics. 2011.
31. JURADO, Carlos Blé. Diseño Ágil con TDD. Primera Edición. 2011
32. JYOTI KAR, Nayan. Adopting Agile Methodologies of Software Development. SETLabs Briefings. 2006.
33. KAPPEL Gerti, PROLL Birgit, REICH Siegfried, RETSCHITZEGGER Werner. Web Engineering: The Discipline of Systematic Development of Web Applications. John Wiley and Sons. 2003.
34. KENDALL Kenneth & KENDALL Julie. Análisis y Diseño de Sistemas de Información. Tercera Edición. 2005.
35. KNIBERG Henrik. Scrum y XP desde las trincheras. InfoQ. 2007.
36. LAM, Monica. Methodologies, tools, and techniques in practice for Web Application Development. California State University, Journal of Technology Research. 2012.
37. LEFF , Avraham, RAYFIELD, James. Web-Application Development Using the Model View Controller Design Pattern. Enterprise Distributed Object Computing Conference. 2001.
38. LEÓN SERRANO Gonzalo. Ingeniería de Sistemas de Software. Primera Edición. Isdefe. 1997
39. MAC IVER Robbie. Scrum Is Not Just for Software. ScrumAlliance. 2009.
40. MARTIN , Robert C.. Clean Code – A Handbook of Agile Software. Pearson Education. 2009.
41. MARTÍNEZ Esther. E-Business para las PYMES. 2003.
42. MENDOZA, M., BARRIOS, J.. Propuestas metodológicas para el desarrollo de aplicaciones Web: una evaluación según la ingeniería de métodos. Universidad de los Andes, Venezuela. 2004.
43. MONTAÑA ORTEGA Juan Sebastián. Methodologies for Developing Web Applications and their Relation with MDA Approach. Universidad de los Andes, Colombia. 2008.

- 44.MUSHTAQ, Zaigham, M. QURESHI, Rizwan Jameel. Novel Hybrid Model: Integrating Scrum and XP. MECS. 2012.
- 45.NIETO MELGAREJO Patricia. Nociones generales sobre el comercio electrónico. Lima-Perú, 2005.
- 46.OLIVEROS, Alejandro; WEHBE, Ricardo; DEL VALLE ROJO, Silvana; ROUSSELOT, Julián. Requerimientos para aplicaciones web. XIII Workshop de Investigadores en Ciencias de la Computación. Universidad Nacional de La Plata. 2011.
- 47.OLTRA Juan V. Clasificaciones del comercio electrónico. 2003.
- 48.PALACIO, Juan, RUATA Claudia. Scrum Manager - Gestión de Proyectos. ScrumManager. 2011.
- 49.PALACIO, Juan. Scrum Manager - Manual - Gestión de Proyectos. ScrumManager. 2008.
- 50.PHAN, Andrew, PHAM. Phuong-Van Scrum in Action: Agile Software Project Management and Development. Cengage Learning. 2012.
- 51.PICHLER Roman. Agile Product Management with Scrum. Addison Wesley. 2010.
- 52.PINEDA MORENO, Milena. Estudio de factibilidad sobre el desarrollo de software a la medida de marcación predictiva. Colombia. 2007.
- 53.PRESSMAN, Roger. Ingeniería del Software. Un Enfoque Práctico. Sexta Edición. McGraw-Hill. Madrid. 2005.
- 54.Puromarketing.com. Internet y e-commerce: las ventajas de las tiendas online para su negocio. 2012.
- 55.Rana TASSABEHJI. Understanding E-commerce for Business. Sage Publications. 2003.
- 56.Raúl HERRANZ. Scrum Manager - En busca de la excelencia del código. ScrumManager. 2012.
- 57.SANDERS , William. PHP Design Patterns. O'Reilly. 2013.
- 58.SCHACH Stephen. Object-Oriented and Classical Software Engineering. Octava Edición. McGraw-Hill. 2010.
- 59.SCHWABER Ken, SUTHERLAND Jeff. The Scrum Guide. 2011.
- 60.SCHWABER, Ken. Agile Project Management with Scrum. Microsoft Press. 2004.

61. SHORE James, WARDEN Shane. The Art of Agile Development. O'Reilly. 2007.
62. SINHA , Rohit. Software Development with Agile Approach. PMI Virtual Library. 2010.
63. SOMMERVILLE Ian. Ingeniería del Software. Séptima Edición. Prentice Hall. 2005.
64. SRIDARAN R., PADMAVATHI G., IYAKUTTI K. A Survey of Design Pattern Based Web Applications. Chair of Software Engineering. 2009.
65. SWEAT, Jason E.. PHP Architect Guide to PHP Design Patterns. Marco Tabini & Associates. 2005.
66. TAMRAKAR, Ritesh, JORGENSEN, Magne. Does the Use of Fibonacci Numbers in Planning Poker Affect Effort Estimates?. Simula Research Laboratory. 2012.
67. TAYLOR, M.J., MCWILLIAM, J., FORSYTH , H., WADE S.. Methodologies and website development: a survey of practice. School of Computing and Mathematical Sciences. Liverpool John Moores University. 2002.
68. TINOCO Oscar, ROSALES Pedro Pablo, SALAS Julio. Criterios de selección de metodologías de desarrollo de software. Revista de la Facultad de Ingeniería Industrial – Universidad Nacional Mayor de San Marcos. 2010.
69. TREJO DELARBRE Raúl. Vivir en la Sociedad de la Información. Revista Iberoamericana de Ciencia, Tecnología, Sociedad e Innovación. Número 1 – Septiembre. 2001.
70. WAKE , William. Extreme Programming Explored. 2000.
71. WYSOCKI , Robert. Effective Project Management - Traditional, Agile, Extreme. Wiley Publishing. 2009.

DIRECCIONES ELECTRONICAS

72. Agile Alliance. Daily Meeting.
<<http://guide.agilealliance.org/guide/daily.html>>

73. Alexandre Magno. The Daily Meeting Trap. ScrumAlliance.
<<http://www.scrumalliance.org/articles/62-the-daily-meeting-trap->. 2007>
74. Anand Gothe. Importance of Product Backlog Refinement in Scrum.
<<http://www.prowareness.com/blog/importance-of-product-backlog-refinement-in-scrum/>. 2010>
75. Asociación Española de la Economía Digital (adigital). Libro blanco del comercio electrónico. 2012. <<http://www.libroblanco.adigital.org/>>
76. Don Wells. Simplicity is the Key.
<<http://www.extremeprogramming.org/rules/simple.html>. 2009. José Joskowicz. Reglas y Prácticas de Extreme Programming. Universidad de Vigo. 2008>
77. Dusan Kocurek. Understanding the Scrum Burndown Chart. ScrumDesk.
<<http://www.methodsandtools.com/archive/scrumburndown.php>. 2011>
78. Hosting Review. Mejores compañías de Hosting. <http://www.hosting-review.com/us-es.shtml?gclid=CKLhslqzzbcCFSgS7Aod7jgA_w>
79. El estudio del comercio electrónico en América Latina 2012.
<<http://especiales.americaeconomia.com/2012/comercio-electronico-america-latina-2012/estudio.php>>
80. International Scrum Institute. Scrum Burndown Chart. <http://www.scrum-institute.org/Burndown_Chart.php. 2013>
81. Jason Mark. Agile Web Development That Works. sixrevisions.com
<<http://sixrevisions.com/web-development/agile/>. 2011>
82. Jeff McKenna. Does Agile Development Require UML?. Jeff McKenna's Agile Development Blog. <<http://blog.agile-action.com/uml/>. 2009>
83. Juan Quijano. Cuando todos son ventajas, Sprint Backlog – Hablando de Scrum. genbetadev.com. <<http://www.genbetadev.com/metodologias-de-programacion/cuando-todos-son-ventajas-sprint-backlog-hablando-de-scrum>. 2012>
84. Laszlo Szalvay. Scrum Sprint. Scrum Methodology – CollabNet.
<<http://scrummethodology.com/scrum-sprint/>. 2010>
85. Luxor Technologies. B2C Business to Consumer.
<<http://www.luxortec.com/preguntas-frecuentes/b2c-business-to-consumer/>. 2011>

86. Markus Sprunck. Product-Burndown-Charts and Sprint-Burndown-Charts in SCRUM Projects. <<http://www.javacodegeeks.com/2012/06/product-burndown-charts-and-sprint.html>. 2012>
87. Martin Fowler. Continuous Integration.
<<http://www.martinfowler.com/articles/continuousIntegration.html>. 2006>
88. Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ken Schwaber, entre otros. The Agile Manifesto. Principios del Manifiesto Ágil.
<<http://agilemanifesto.org/iso/es/principles.html>. 2001>
89. Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ken Schwaber, entre otros. The Agile Manifesto. Valores del Manifiesto Ágil.
<<http://agilemanifesto.org/iso/es/>. 2001>
90. Mike Cohn. Agile Burn Chart - Release Burndown Charts. Mountain Goat Software. <<http://www.mountaingoatsoftware.com/scrum/release-burndown>. 2012>
91. Monjurul Habib. Agile software development methodologies and how to apply them. codeproject.com
<<http://www.codeproject.com/Articles/604417/Agile-software-development-methodologies-and-how-t>. 2014>
92. ONTSI. Comercio Electrónico B2C 2011. España. 2012.
<<http://www.ontsi.red.es/ontsi/es/estudios-informes/estudio-b2c-2011-edicion-2012>>
93. Proyectos Ágiles. Ejecución de la iteración (Sprint).
<<http://www.proyectosagiles.org/ejecucion-iteracion-sprint>. 2010>
94. Proyectos Ágiles. Ejemplo de uso del tablero o pizarra de tareas (Scrum Taskboard). <<http://www.proyectosagiles.org/ejemplo-tablero-pizarra-tareas-scrum-taskboard>. 2010>
95. Rini van Solingen. How to improve your Sprint Retrospective meeting?. <<http://rinivansolingen.nl/how-to-improve-your-sprint-retrospective-meeting/?lang=en>. 2013>
96. Sathees Kumar. What is Agile Scrum?.
<<http://satheespractice.blogspot.com/2011/11/what-is-agile-scrum.html>. 2011>
97. Scrum for Team System. Sprint Review.

- <<http://www.scrumforteamssystem.co.uk/ProcessGuidance/scrum/process/sprint-review>. 2010>
98. ScrumSense. What every Product Owner should know. 2011. <<http://www.scrumsense.com/wp-content/uploads/2011/02/Product-Owners-Manual.pdf>>
99. Sonja Rosúa Clyne. Cómo funciona el modelo de comercio electrónico C2C (Consumer to Consumer). <<http://thefik.com/es/post/how-the-consumer-to-consumer-c2c-e-commerce-model-works/>. 2013>
100. Tara Hamilton-Whitaker. Scrum Sprint Planning Meetings – Who, What, When, Where, Why. <<http://agile101.net/2009/07/24/scrum-sprint-planning-meetings-who-what-when-where-why/>. 2010>
101. Tony Marston. The Model-View-Controller (MVC) Design Pattern for PHP. <http://www.tonymarston.net/php-mysql/model-view-controller.html>. 2013
102. Zend Framework para PHP. Estándares de Codificación. <http://doczf.mikaelkael.fr/1.10/es/coding-standard.html>

ANEXOS

1. Formatos de Acta de Reunión.
2. Manual de Instalación (Anexo Digital).
3. Manual de Usuario (Anexo Digital)

Lugar, fecha

ACTA DE REUNIÓN DAILY MEETING

El daily meeting correspondiente a la fecha se resume en la siguiente tabla:

Participante	¿Qué ha hecho desde la última reunión?	¿Qué planea hacer desde este momento?	¿Qué impedimentos tuvo o va a tener para cumplir con los compromisos en esta iteración y en el proyecto?
Participante 1			
Participante 2			
.....			

FIRMA DE LOS PARTICIPANTES		
_____	_____	_____
Participante1	Participante 2	Participante

4.2.1.1 Lugar, fecha

ACTA DE REUNIÓN SPRINT PLANNING MEETING

Conforme a lo expresado durante la el Sprint Planning Meeting de la fecha, que da apertura al Sprint # , el product Owner, en base a las consideraciones mencionadas por los demás miembros del equipo, ha decidido las historias de usuario que conforman esta iteración.

Por su parte cada miembro que considerò estar en la capacidad de tomar parte en las tareas detalladas ha colocado su nombre en el tablero, quedando el siguiente Sprint Backlog resultante:

1HISTORIA DE USUARIO	TAREAS	TIEMPO (horas)	TAREA TOMADA POR:
Nombre Historia Usuario	Tarea 1	X	Participante 1
	Tarea 2	X	Participante 1
	Tarea 3	X	Participante 2
	X	Participante 3

FIRMA DE LOS PARTICIPANTES		
_____	_____	_____
Participante 1	Participante 2	Participante

Fecha, lugar

ACTA DE REUNIÓN PRODUCT BACKLOG REFINEMENT

Al dar por concluido el Sprint # se ha realizado una revisión de las historias de usuario pendientes en el Product Backlog, Tras el aporte de ideas dada por cada miembro del equipo, el product Owner decidió que SI/NO se aumenten las siguientes historias de usuario al Product Backlog existente.

1HISTORIA DE USUARIO	SPRINT TENTATIVO	TIEMPO (DIAS)
Nombre Historia Usuario	X	Y
Nombre Historia Usuario	A	B

FIRMA DEL PRODUCT OWNER

Fecha, lugar

ACTA DE REUNIÓN SPRINT RETROSPECTIVE

Al finalizar el Sprint # y tras la reunión del equipo, se determinaron los factores que influyeron en la no conclusión de las tareas planeadas.

Los aportes se resumen en la siguiente tabla:

PARTICIPANTE	OBSERVACIÓN
Participante 1	Se subestimó el tiempo necesario para llevar a cabo la tarea X
Participante 1	No se tomó en cuenta la necesidad de implementar los métodos X, Y,Z, para conseguir la funcionalidad del módulo A.

Con todas las aportaciones se tomará las medidas necesarias para no incurrir en los mismos errores en el Sprint siguiente.

FIRMA DE LOS PARTICIPANTES		
_____	_____	_____
Participante1	Participante 2	Participante

Fecha, lugar

ACTA DE REUNIÓN SPRINT REVIEW

Al finalizar el Sprint # y tras la reunión del equipo en una reunión que tomó un tiempo de x horas, se realizó la presentación de las metas cumplidas en esta iteración.

Las tareas que llegaron a concluir como se esperaba se listan a continuación:

TAREAS CUMPLIDAS	PARTICIPANTE
Tarea X	Participante 1
Tarea Y	Participante 1

Todos los presentes mostramos la conformidad con la funcionalidad alcanzada en las tareas antes mencionadas.

Expresamos nuestro acuerdo con nuestras firmas.

<i>FIRMA DE LOS PARTICIPANTES</i>		
_____	_____	_____
Participante1	Participante 2	Participante