PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS



DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS Y COMPUTACIÓN

DESARROLLO DE UN SOFTWARE PROTOTIPO PARA LA ADMINISTRACIÓN DE UNA RED DE MERCADEO, CON UN SISTEMA DE COMPENSACIÓN UNINIVEL SIMPLE Y COMPACTACIÓN DE SOCIOS DISTRIBUIDORES

AUTOR:
DOUGLAS EFRAIN JARA NEGRETE

DIRECTOR:
ING. BEATRIZ CAMPOS

QUITO, MAYO 2017

A mi madre.

AGRADECIMIENTOS

A mi esposa Randy, por acompañarme durante este largo trayecto y a mis hijos, Dante y Damián, por ser un gran apoyo siempre.

A mis padres Marilena y Medardo por nunca dejar que me rinda, y por tantas enseñanzas y lecciones de vida.

A mis hermanos, Eliza y David, por demostrarme lo lejos que se puede llegar.

A todos los docentes de la Facultad de Ingeniería, Escuela de Sistemas de la PUCE por la gran labor de compartir su conocimiento.

A mi directora y correctores, Beatriz Campos, Javier Cóndor y Eddie Sánchez, por su valioso tiempo y apoyo.

INDICE DE CONTENIDOS

TEMA I

ASPECTOS PRELIMINARES O INTRODUCTORIOS II

DEDICATORIA II

INDICE DE CONTENIDOS IV

INDICE DE TABLAS IX

INDICE DE ILUSTRACIONES XI

CUERPO DEL TRABAJO 12

RESUMEN 12

CAPÍTULO 1. INTRODUCCIÓN 13

1.1. Tema 13

1.2. Justificación 13

1.3. Objetivos 14

1.3.1. Objetivo General 14

1.3.2. Objetivos Específicos 14

1.4. Información General de las Redes de Mercadeo 14

1.4.1. Modelo de Negocio 14

1.4.2. Redes de Mercadeo 15

1.4.3. Aspectos Legales 15

1.4.4. Historia 19

1.4.5. Sistemas de Compensación 21

1.4.6. MLM en Ecuador 23

CAPÍTULO 2. MARCO TEÓRICO 27

2.1. Metodología de Desarrollo XP 27

2.1.1. Conceptos 27

2.1.2. Modelo XP 29

2.2. Software Libre 32

2.2.1. Las cuatro libertades esenciales 32

2.3. XAMPP 32

2.4. Lenguage HTML 33

2.4.1. Definición 33

2.4.2. Historia Del HTML 33

2.4.3. Etiquetas HTML 34

2.4.4. HTML 5 35

2.5. CSS 36

2.6. PHP 37

2.6.1. Historia 38

2.7. MySQL 39

2.7.1. Características 39

2.7.2. Licencia 40

2.8. AJAX y Javascript 40

2.8.1. Uso de AJAX en la programación web 41

2.9. CodeIgniter: Framework de desarrollo 42

2.9.1. Definición del término “framework” 42

2.9.2. CodeIgniter 42

2.9.3. Características generales.....42

2.9.4. Breve historia.....43

2.9.5. Pasos para la Instalación 43

2.9.6. Flujo de la Aplicación.....45

2.9.7. MVC (Modelo-Vista-Controlador).....46

CAPÍTULO 3. ANÁLISIS DE REQUERIMIENTOS.....47

3.1. Definiciones.....47

3.2. Requerimientos no funcionales 48

3.3. Requerimientos Funcionales 48

3.3.1. Resumen de las entrevistas.....49

3.3.2. Historias de usuarios 54

3.3.3. Resumen de requerimientos funcionales 59

CAPÍTULO 4. DISEÑO DEL PROTOTIPO 60

4.1. Diseño de la base de datos 60

4.1.1. Diseño conceptual 60

4.1.2. Diseño lógico 61

4.1.3. Diseño físico 62

4.2. Diseños del prototipo 68

4.2.1. Arquitectura del software 68

4.2.2. Diagramas de secuencia 68

4.3. Diseño de interfaces 73

4.3.1. Login 73

4.3.2. Dashboard 74

4.3.3. Red 74

4.3.4. Socios 75

4.3.5. Items 76

4.3.6. Facturación 77

4.3.7. Parámetros 77

CAPÍTULO 5. IMPLEMENTACIÓN Y PRUEBAS 79

5.1. Plan de iteraciones 79

5.2. Pruebas del prototipo 79

5.2.1. Gestionar socios 80

5.2.2. Gestionar y obtener red de un socio 81

5.2.3. Obtener comisiones de un socio 81

5.2.4. Obtener comprobante de pago 82

5.2.5. Gestionar ítems 83

5.2.6. Gestionar formas de pago 84

5.2.7. Gestionar países 85

5.2.8. Gestionar configuraciones del sistema 86

5.2.10. Obtener estadísticas 87

CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES 88

6.1. Conclusiones 88

6.2. Recomendaciones 89

GLOSARIO DE TÉRMINOS 91

REFERENCIAS BIBLIOGRÁFICAS 94

ANEXOS 99

INDICE DE TABLAS

Tabla 1-1: Diferencia entre un sistema multinivel y las nefastas pirámides. .19
Tabla 2-1: Versiones vs. Años de publicación del HTML (World Wide Web Consortium, 2015) 34
Tabla 3-1: Historia de usuario No. 1 – Gestionar socios. 54
Tabla 3-2: Historia de usuario No. 2 – Gestionar red de socios. 54
Tabla 3-3: Historia de usuario No. 3 – Obtener red de un socio. 55
Tabla 3-4: Historia de usuario No. 4 – Obtener comisiones de un socio. 55
Tabla 3-4: Historia de usuario No. 5 – Obtener comprobante de pago. 56
Tabla 3-5: Historia de usuario No. 6 – Gestionar ítems. 56
Tabla 3-7: Historia de usuario No. 7 – Gestionar formas de pago. 57
Tabla 3-8: Historia de usuario No. 8 – Gestionar países 57
Tabla 3-9: Historia de usuario No. 9 – Gestionar configuraciones del sistema. 58
Tabla 3-10: Historia de usuario No. 10 – Obtener estadísticas. 58
Tabla 3-11: Resumen de Requerimientos 59
Tabla 4-1: Tabla clientes 62
Tabla 4-2: Tabla comisiones 62
Tabla 4-3: Tabla comisiones_facturas_forma_pagos 63
Tabla 4-4: Tabla configuraciones 63
Tabla 4-5: Tabla configuraciones 64
Tabla 4-6: Tabla facturas_forma_pago 64
Tabla 4-7: Tabla facturas_items 64
Tabla 4-8: Tabla facturas_socios 65
Tabla 4-9: Tabla forma_pagos 65
Tabla 4-1: Tabla Ítems 65
Tabla 4-11: Tabla paises 66
Tabla 4-12: Tabla socios 66
Tabla 4-13: Tabla socios_padres_log 66
Tabla 4-14: Tabla users 67

Tabla 4-15: Tabla users_groups 67
Tabla 4-16: Tabla groups 67
Tabla 5-1: Planificación de iteraciones 79
Tabla 5-2: Pruebas unitarias para la gestión de socios 80
Tabla 5-3: Pruebas unitarias para la gestión y obtención de la red de un socio 81
Tabla 5-4: Pruebas unitarias para la obtención de las comisiones de un socio 81
Tabla 5-5: Pruebas unitarias para obtener un comprobante de pago 82
Tabla 5-6: Pruebas unitarias para la gestión de ítems 83
Tabla 5-7: Pruebas unitarias para la gestión de formas de pago 84
Tabla 5-8: Pruebas unitarias para la gestión de países 85
Tabla 5-9: Pruebas unitarias para la gestión de configuraciones del sistema 86
Tabla 5-10: Pruebas unitarias para la obtención de estadísticas 87

INDICE DE ILUSTRACIONES

Ilustración 2-1: Comparación de fases entre otras metodologías y XP. 30
Ilustración 2-2: Ejemplo de codificación en el lenguaje PHP. 37
Ilustración 2-3: Flujo de una Aplicación en CodeIgniter 45
Ilustración 3-1: Estructura básica de una historia de usuario 48
Ilustración 3-1: Sistema de compensación uninivel con compactación simple 50
Ilustración 3-2: Sistema de compensación uninivel con compactación simple – Jerarquía 51
Ilustración 3-3: Sistema de compensación uninivel con compactación simple - Compactación 52
Ilustración 4-1: Esquema conceptual del prototipo 60
Ilustración 4-2: Diseño lógico del prototipo 61
Ilustración 4-3: Flujo de una Aplicación en CodeIgniter 68
Ilustración 4-4: Diagrama de secuencia para la creación de elementos 69
Ilustración 4-5: Diagrama de secuencia para la modificación de elementos. 70
Ilustración 4-6: Diagrama de secuencia para la eliminación de elementos... 71
Ilustración 4-7: Diagrama de secuencia para la facturación 72
Ilustración 4-8: Interface de Ingreso 73
Ilustración 4-9: Interface del Dashboard 74
Ilustración 4-10: Interface de la Red de un Socio 75
Ilustración 4-11: Interface Socios 76
Ilustración 4-12: Interface de Items 76
Ilustración 4-13: Interface de Facturación 77
Ilustración 4-14: Interface de Parámetros 78

RESUMEN

Este documento presenta el desarrollo de un software prototipo que resuelva los requerimientos de una organización que desee implementar redes de mercadeo en su modelo de negocio, usando un sistema de compensación uninivel simple con compactación de socios. Además, presenta las herramientas que se usan para construir el mencionado prototipo de software.

Para que el desarrollo tenga un diseño sólido y modular se usa el patrón de diseño MVC (Modelo-Vista-Controlador). Como lenguaje de programación del lado del servidor se usa PHP, del lado del cliente Javascript, y como complementos para la presentación y experiencia del usuario se usa CSS, HTML y AJAX.

El software desarrollado cubre los requerimientos para la correcta administración de la red de socios de una organización que use (o pretenda usar) el mencionado sistema de compensación.

Elegir como método de levantamiento de requerimientos a la entrevista fue clave para poder comprender el funcionamiento de una red con estas características y así plasmar dichas ideas en la construcción del sistema web.

La óptima combinación de tecnologías usadas para la construcción del software, y la correcta implementación de las entrevistas para el levantamiento de requerimientos, contribuyeron para obtener el resultado deseado en el software desarrollado.

CAPÍTULO 1. INTRODUCCIÓN

1.1. Tema

Desarrollo de un software prototipo para la administración de una red de mercadeo, con un sistema de compensación uninivel simple y con compactación de socios.

1.2. Justificación

Actualmente en el Ecuador existen varias empresas como Yanbal, Omnilife, Leonisa, Avon, Herbalife, entre otras, que tienen implementando como modelo de negocios a las redes de mercadeo para impulsar sus ventas. Estas empresas son, en su mayoría, franquicias que desde sus inicios manejan este tipo de métodos con diferentes sistemas de compensación.

Según Direct Selling News[1], a nivel mundial, está creciendo el número de empresas que brindan consultorías para la implementación de un sistema de redes de mercadeo en organizaciones que usan el sistema de ventas tradicional ya que, con el pasar del tiempo, la dificultad para vender a los consumidores ha crecido considerablemente, y el “Network Marketing” da a las empresas más acceso al mercado -comparado con el sistema tradicional- ya que se basa en la recomendación “boca a boca” de sus consumidores, quienes obtienen un bono o comisión por esa recomendación.

Localmente surge, en las pequeñas y medianas empresas, la necesidad de implementar en su modelo de negocio un paradigma que les ayude a aumentar su acceso al mercado. Este proyecto de disertación sirve de herramienta tecnológica complementaria para resolver esta necesidad.

[1] Artículo en http://marketingdered.com/la-diferencia-del-1-en-las-ventas-directas/ que toma como fuente Direct Selling News.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un software prototipo que administre una red de mercadeo con un sistema de compensación uninivel simple y compactación de socios, en plataforma web, utilizando software libre.

1.3.2. Objetivos Específicos

- Definir los requerimientos para el desarrollo del software prototipo que administre una red de mercadeo con un sistema de compensación uninivel simple y con compactación de socios.
- Realizar el diseño lógico y físico del prototipo.
- Desarrollar el sistema prototipo para un ambiente web.
- Realizar pruebas y verificación de requerimientos.
- Realizar conclusiones y recomendaciones.

1.4. Información General de las Redes de Mercadeo

1.4.1. Modelo de Negocio

También llamado diseño de negocio o diseño empresarial, es el mecanismo por el cual un negocio trata de generar ingresos y beneficios. Es un resumen de cómo una compañía planifica servir a sus clientes. Implica tanto el concepto de estrategia como el de implementación.

1.4.2. Redes de Mercadeo

Las redes de mercadeo, marketing multinivel o sistemas de mercadeo en red, son un modelo de negocio en el cual, la estrategia de la empresa para obtener ingresos o beneficios es captar más distribuidores de su(s) producto(s), creando así una red de personas, organizada por niveles según el esquema que maneje la empresa.

Cada persona que se asocia con la compañía recibe una compensación en base a sus ventas y comisiones generadas por la red de nuevos socios creada por esta persona.

Para ser un socio o representante de este tipo de empresas, se debe realizar una inversión de dinero, y/o adquirir cierta cantidad de productos. Es decir, lo que adquiere el nuevo socio es un sistema comprobado de negocio, similar a una franquicia, pero con una inversión relativamente baja.

Robert Kiyosaki, en su libro “El cuadrante del flujo del dinero”, menciona que:

“Debido a los avances tecnológicos en la industria de las computadoras, estas organizaciones están totalmente automatizadas, y los dolores de cabeza del papeleo, el procesamiento de pedidos, la distribución, la contabilidad y el seguimiento son manejados casi por completo por sistemas de programas de computadoras de mercadeo en red” (Kiyosaki, 2004).

1.4.3. Aspectos Legales

Antes de adquirir una representación en una empresa de marketing multinivel o involucrarse activamente en una, se debe tener en cuenta ciertos factores importantes en cuanto al manejo de la organización para asegurarse de que cumpla con todos los aspectos legales requeridos.

Por ello, la Comisión Federal de Comercio de USA (2015) hace una serie de advertencias para el público que está interesado en ingresar a una red de mercadeo o a un nuevo esquema de negocios:

“No todos los planes de comercialización multiniveles son legítimos. Si el dinero que gana se basa en lo que le vende al público, es un plan de comercialización multiniveles. Si lo que usted gana se basa en la cantidad de personas que recluta y en lo que les venda a esas personas, es un esquema piramidal. Los esquemas piramidales son ilegales, y la inmensa mayoría de los participantes pierden dinero.” (Comisión Federal de Comercio de USA, 2015)

La Asociación Ecuatoriana de Venta Directa, expone las diferencias entre un sistema piramidal (ilegal), y un sistema multinivel o de mercadeo en red usando la siguiente tabla comparativa:

MULTINIVEL	PIRAMIDAL
Pequeña cuota de ingreso La cantidad que exigen es siempre relativamente baja. Dicha cantidad cubre principalmente el costo de los productos de demostración y material de entrenamiento y de ayuda para la venta. Sólo en caso de equipos, el monto puede ser alto, pero está relacionado con el equipo que se ofrece.	**Cuota de ingreso elevada** Camuflan a menudo sus altas cuotas de ingreso como parte del importe del material de entrenamiento, stock del producto, etc. El grueso de los beneficios de los esquemas piramidales proviene del reclutamiento de personas, no de la venta de productos.

MULTINIVEL	PIRAMIDAL
Productos de calidad Venden al público en general una amplia gama de productos de calidad. Estimulan la compra repetida de productos por parte de los consumidores satisfechos.	**Productos de dudosa calidad** No demuestran interés alguno en comercializar sus productos, generalmente de dudoso valor. El dinero se consigue mediante el típico sistema piramidal de "reclutamiento". El producto es secundario, lo importante es el ingreso de personas a la pirámide.
Garantía de devolución Se garantiza la devolución de los artículos no vendidos. Si el cliente no está satisfecho con el producto lo puede devolver, salvo ciertas excepciones, por ejemplo, ropa interior.	**Sin garantía** No aseguran la recompra de las existencias no vendidas. No hay nada que devolver.
Un negocio a largo plazo Las compañías legítimas de venta directa están interesadas en el negocio a largo plazo. Esto supone que deben establecer relaciones sólidas mercantiles con sus distribuidores independientes, microempresarios.	**"Hágase rico en poco tiempo"** Son engañosos métodos de enriquecimiento rápido. El elevado número de personas que componen la base pagan dinero al reducido grupo que está en lo alto. Claramente, el sistema no puede sostenerse durante mucho tiempo.

MULTINIVEL	PIRAMIDAL
Reconocimiento del esfuerzo El éxito depende del esfuerzo. Eso supone que los ingresos de un distribuidor independiente son proporcionales a la entrega, esfuerzo y dedicación para con su negocio propio.	**El esfuerzo no significa nada** Se puede comprar una posición concreta en la pirámide. Mientras más dinero invierte, mejor es su posición y mayor el riesgo de ser estafado.
Reducido o nulo stock personal Las compañías legales establecen estrictos Códigos de Conducta que, entre otras cosas, desaconsejan a sus distribuidores cargar con más existencias de las realmente necesarias para su actividad. No se promueve el mantener inventario de productos, no *sobre-estoquearse*.	**Hay que acumular producto** La venta piramidal obliga a cargar con grandes cantidades de producto o bien abonar altas cuotas de ingreso para poder participar.

MULTINIVEL	PIRAMIDAL
Legal La venta directa es un popular medio de comercialización reconocido como legal y legítimo en la mayoría de los países. Las empresas de venta directa miembros de la AEVD, cumplen a cabalidad con las leyes nacionales e internacionales y esto garantiza la idoneidad de la empresa y sus marcas, al consumidor final y a los distribuidores independientes.	**Ilegal u objetable** Los promotores de un esquema piramidal cometen fraude al engañar deliberadamente a los participantes en el mismo. Los esquemas piramidales han sido declarados ilegales en muchos países del mundo. Son empresas que no cumplen con pago de impuestos al SRI. Perjudican al Estado y a toda la Comunidad.

Tabla 1-1: Diferencia entre un sistema multinivel y las nefastas pirámides.
Fuente: (Asociación Ecuatoriana de Venta Directa, AEVD, 2015).

1.4.4. Historia

En la década de los años 20, Carl Rehnborg, un estadounidense que vivía en China, fue tomado prisionero durante varios disturbios políticos ocurridos en Shangai. Durante ese periodo de aislamiento, Carl complementó su escasa dieta, con sopas de hierbas locales, verduras y clavos oxidados (por el hierro). Meses más tarde, él y otros compañeros con los que había compartido esta desagradable, pero saludable mezcla, salieron de la prisión mucho más saludables en comparación con otros cautivos que comieron solamente las raciones otorgadas por el ejército de ese país (NUTRILITE, 2015).

Durante su estadía en China, Carl pudo observar que en las comunidades donde se alimentaban mayormente de frutas, hierbas y verduras, la gente era más saludable que la que se alimentaba de azúcar, sal y grasas.

A su regreso a Estados Unidos en 1934, montó un laboratorio en donde llevó a cabo varios experimentos con plantas y hierbas durante 6 años, además de desarrollar un método para secar las plantas y conservar sus nutrientes importantes. Creó una formulación de alfalfa, berro y perejil, que se convirtió en la base de su primer producto (NUTRILITE, 2015).

California Vitamins fue el nombre de la empresa con la que Rehnborg inició sus actividades comerciales vendiendo suplementos nutricionales, “puerta a puerta”, lo se conoce ahora como venta directa. Se dio cuenta que sus compradores, lo referían con otros (boca a boca), y algunos de sus clientes se convertían en vendedores ya que, además de recibir un ingreso extra, se convertían en dueños de su propio negocio, trabajando, en la mayoría de los casos, a medio tiempo desde su hogar (Samartano, 2015).

En 1941, la empresa cambia de nombre a Nutrilite Products Inc. A Rehnborg no le bastó con el gran éxito de su empresa de multivitamínicos, e inició el diseño de un nuevo y revolucionario esquema de distribución. En 1945, empezó a motivar a sus vendedores permitiéndoles reclutar nuevos distribuidores, y ganar una comisión de las ventas de estos distribuidores. El plan permitía a cualquier distribuidor de Nutrilite con 25 clientes minoristas regulares, asociar nuevos vendedores (distribuidores) y obtener una comisión del 3% por sus ventas. Esta comisión se mantenía mientras el asociado permaneciera en Nutrilite, así, por primera vez los vendedores directos comunes podían desarrollar una organización de ventas que generara ingresos residuales. Una de las ventajas que la gente percibía de este novedoso esquema era que no se requería ser un erudito en finanzas, administración o ventas para lograr sacar provecho de la empresa, ya que el sistema se encargaría de todos los detalles administrativos (Diaz M., 2010).

A partir de 1950, las empresas empiezan a adoptar este sistema de ventas en su negocio, y algunas nuevas empresas lo toman con el núcleo de su sistema de ventas. En 1954, nace una empresa llamada Amway, abreviación de “American Way of Life”. Fundada por 2 antiguos distribuidores de Nutrilite,

Rich DeVos y Jay Van Andel, Amway crece de tal modo que absorbe a Nutrilite, y pasa a ser su división nutricional (Samartano, 2015).

En estos días Amway es la más grande firma a nivel mundial, con distribución en más de 80 países y con más de 3 millones de distribuidores. Tiene ventas anuales estimadas de 10.80 billones de dólares. (Direct Selling News, 2015)

El multinivel fuera de los Estados Unidos se expandió y cientos de empresarios entraron a la industria creando sus propias compañías.

Dentro del Ecuador, existen varias empresas que manejan el esquema de marketing multinivel, las cuales están legalmente constituidas, funcionan dentro de la ley y llevan varios años en el mercado. Entre ellas están: 4life, Avon, Cy zone, Esika, Herbalife, L'ebel, Leonisa, Nikken, Omnilife, Oriflame, Rainbow, Royal Prestige, Tupperware, Yanbal.

1.4.5. Sistemas de Compensación

Un sistema o plan de compensación es un esquema de pago de comisiones que usan las empresas de la industria multinivel. Estas empresas, en su mayoría, usan un sistema de compensación modificado, el cual tiene como base uno estándar.

En todos los sistemas existe una ganancia personal por la venta de productos, y otra por captar nuevos socios, es decir, por la creación de una red.

Entre los sistemas o planes de compensación más conocidos tenemos:

1.4.5.1. Plan escalonado

También llamado Breakaway. Es el más antiguo y usado. Este plan requiere más esfuerzo al inicio, pero es el que da los mejores ingresos. Los

distribuidores ganan una comisión de sus ventas personales y de las ventas de sus grupos que hay por debajo de ellos, hasta que los igualen en el escalón (Wikipedia, 2015). Se consideran dos tipos de socios: directos y subordinados con tres tipos de remuneración:

Comisiones de base: Como en cualquier organización de ventas, está la comisión que se gana por la operación de los subordinados. Para convertirse en director se debe lograr cierto volumen de ventas grupal.

Comisiones generacionales: Son comisiones por la operación de los grupos de otros directores que eran subordinados y ahora son directores. La mayoría de los planes compensan al menos a tres generaciones de tales directores.

Bonos ejecutivos: Se paga una comisión adicional a los directores que exceden su objetivo de ventas. Por ejemplo, un porcentaje del total de los ingresos de la compañía va a una bolsa de bonos que se reparte mensualmente prorrateando entre los directores que hayan excedido cierto nivel de ventas (Wikipedia, 2015).

1.4.5.2. Plan Matricial

Este plan limita la anchura de cada nivel en un grupo de distribuidores, forzando de este modo a los distribuidores fuertes a apilar a sus nuevos reclutados debajo de gente que no hizo el trabajo de patrocinarlos (Wikipedia, 2015).

1.4.5.3. Plan Binario

En este plan, cada nivel tiene solamente dos socios. Tiene la misma desventaja del matricial y, además, las comisiones están basadas en "ciclos" en los que un distribuidor recibe una cantidad siempre y cuando ambas

patas (socios) consigan cada una un cierto volumen de ventas (Wikipedia, 2015).

1.4.5.4. Estructura ascensor o piramidal

Su legalidad es cuestionada y consiste en un "tablero de juegos" en el que cada distribuidor paga una o más unidades de productos para poder participar o recibir bonificación. Este esquema, es considerado un fraude, ya que se entrega el dinero al fundador del negocio y deja en la quiebra a los últimos que participan (Wikipedia, 2015).

1.4.5.5. Venta Directa

Es uno de los más comunes en el Ecuador. El esquema de remuneración plantea comisiones por venta de productos y bonos por nuevos socios, así como también comisiones por las ventas de los nuevos socios.

1.4.6. MLM en Ecuador

En Ecuador, existen un número considerable de empresas que tienen implementado el esquema multinivel. No todas pertenecen a la AEVD (Asociación Ecuatoriana de Venta Directa). A continuación, se presenta una reseña de las empresas más conocidas a nivel nacional:

AVON

Fundada en 1886, AVON es una de las mayores empresas de venta directa de cosméticos a nivel mundial, con más de 6 millones de representantes, en más de 100 países. Llega a Ecuador en 1992. Como lo expresa su lema “The Company for Women”, la empresa se enfoca netamente en productos para mujeres.

Sitio Web: www.avon.com.ec

Miembro de la AEVD: Si

Plan de compensación: Breakaway uninivel

HERBALIFE

Fundada por Mike Hughes en L.A., California, en 1980. La totalidad de sus productos son nutricionales. A partir del año 2000, se extiende por países de centro y Sudamérica, incluyendo al Ecuador. En la actualidad se encuentra en más de 80 países alrededor del mundo.

Sitio Web: www.herbalife.com.ec

Miembro de la AEVD: Si

Plan de Compensación: Breakaway, 3 niveles

LEONISA

Compañía colombiana dedicada a la venta de ropa interior por catálogo. Nace en 1993, enfocando su negocio a las mujeres latinas, aunque no descarta al sexo masculino como su cliente. Su mercado principal es América Latina, y Ecuador es uno de los países con más movimiento.

Sitio Web: www.leonisa.com

Miembro de la AEVD: Si

Plan de Compensación: Venta Directa

ROYAL PRESTIGE

Bajo el nombre jurídico de Majestic Corporation Cia. Ltda., inicia operaciones en el Ecuador en noviembre del 2003, representando a las marcas estadounidenses Royal Prestige y Ocean Blue. La empresa en USA fue

fundada en 1959 ofreciendo ollas fabricadas con acero quirúrgico, purificadores de aire y aspiradoras.

Sitio Web: www.royalprestige.com.ec

Miembro de la AEVD: No

Plan de Compensación: Breakaway

TUPPERWARE

Brinda sus servicios desde 1946, en el periodo de la post guerra. Fundada por Earl Tupper, y bajo el nombre de Tupper Plastics Company, la empresa ofrece a sus clientes una gran variedad de plásticos contenedores para cocina que tienen como valor agregado su hermetismo. En Ecuador ya tiene 18 años operando en el mercado.

Sitio Web: www.tupperware.com.ec

Miembro de la AEVD: Si

Plan de Compensación: Breakaway

OMNILIFE

En 1991, Jorge Vergara crea la empresa Omnitrition, en México. Diez años después, cambia el nombre a Omnilife de México, ofreciendo productos nutricionales. Años después crea una nueva línea de productos de belleza. Omnilife, está presente en más de 19 países, entre ellos Ecuador.

Sitio Web: www.omnilife.com/ecuador

Miembro de la AEVD: No

Plan de Compensación: Breakaway, 6 niveles

YANBAL

En 1967 inicia sus operaciones con la distribución de cosméticos, y con Fernando Belmont a la cabeza. En 1977 llega al Ecuador, convirtiéndose en la primera empresa con un modelo de mercadeo en red en el país y la región, incluso antes de Colombia y Bolivia. Ofrece productos de belleza en 10 países en América y Europa.

Sitio Web: www.yanbal.com

Miembro de la AEVD: Si

Plan de Compensación: Breakaway

CAPÍTULO 2. MARCO TEÓRICO

En este capítulo se detallará las definiciones de todas las herramientas a utilizar en el desarrollo del software prototipo para la administración de una red de mercadeo con un sistema de compensación uninivel simple con compactación de socios.

2.1. Metodología de Desarrollo XP

2.1.1. Conceptos

Extreme Programming (XP) es un estilo de desarrollo de software que se enfoca en una excelente aplicación de técnicas de programación, una comunicación clara, y un trabajo en equipo para lograr el cumplimiento de tareas que anteriormente no se lograba. XP incluye:

- Una filosofía de desarrollo de software basada en los valores de comunicación, retroalimentación, simplicidad, coraje y respeto.
- Un conjunto de prácticas comprobadas, útiles en el mejoramiento del desarrollo de software, las cuales han sido escogidas como expresiones de sus valores.
- Un conjunto de principios complementarios, técnicas intelectuales para trasladar sus valores a prácticas, útil cuando no existe una práctica para un problema en particular.
- Una comunidad que comparte dichos valores y muchas de sus prácticas.

En la primera edición de "*Extreme Programming Explained: Embrace Change*"[2], se tiene una clara definición de lo que es XP:

"XP es una metodología ligera para equipos de desarrollo medianos y pequeños frente a requerimientos vagos o muy cambiantes."

Durante los siguientes años, a esta definición se le agregaron algunas características:

- XP es ligera. En XP no solo se puede hacer lo que se necesita para crear valor para el cliente. No se puede llevar a cuestas una gran cantidad de equipaje y moverse rápido. Sin embargo, no existe un proceso de software rígido. El conjunto de conocimiento técnico necesario para ser un equipo excepcional es extenso y creciente.
- XP es una metodología basada en abordar las limitaciones en el desarrollo de software. Esto no significa abordar la administración del portafolio del proyecto, su justificación, operaciones, marketing o ventas. XP tiene implicaciones en estas áreas, pero no tiene prácticas para abordarlas directamente. La metodología se interpreta a menudo como "un conjunto de reglas a seguir que garantiza el éxito." Las metodologías no funcionan como programas. La gente no es una computadora. Todo equipo aplica XP de diferente forma y con varios grados de éxito.
- XP puede trabajar con equipos de cualquier tamaño. Años atrás, no se quería abarcar mucho. Desde entonces, otros equipos empezaron a usar XP en un amplio rango de proyectos, y tuvieron éxito con grandes y pequeños equipos en grande y pequeños proyectos. Las prácticas necesitan ser argumentadas y alteradas cuando mucha gente está involucrada.

[2] Beck, Kent – Andres, Cynthia, *"Extreme Programming Explained: Embrace Change",* Primera Edición

- XP se adapta a requerimientos vagos o muy cambiantes. XP es buena para estos casos, lo que es excelente porque los requerimientos necesitan cambiarse y adaptarse rápidamente en el mundo de los negocios de hoy en día. Sin embargo, los equipos también han tenido éxito usando XP en donde los requerimientos no parecían volátiles, como por ejemplo proyectos portables.

(Beck & Andres, 2012)

2.1.2. Modelo XP

El ciclo de vida de un proyecto XP incluye, al igual que las otras metodologías, entender lo que el cliente necesita, estimar el esfuerzo, crear la solución y entregar el producto final al cliente. Sin embargo, XP propone un ciclo de vida dinámico, donde se admite expresamente que, en muchos casos, los clientes no son capaces de especificar sus requerimientos al comienzo de un proyecto. Por esto, se trata de realizar ciclos de desarrollo cortos (llamados iteraciones), con entregables funcionales al finalizar cada ciclo. En cada iteración se realiza un ciclo completo de análisis, diseño, desarrollo y pruebas, pero utilizando un conjunto de reglas y prácticas que caracterizan a XP. (Joskowicz, 2008)

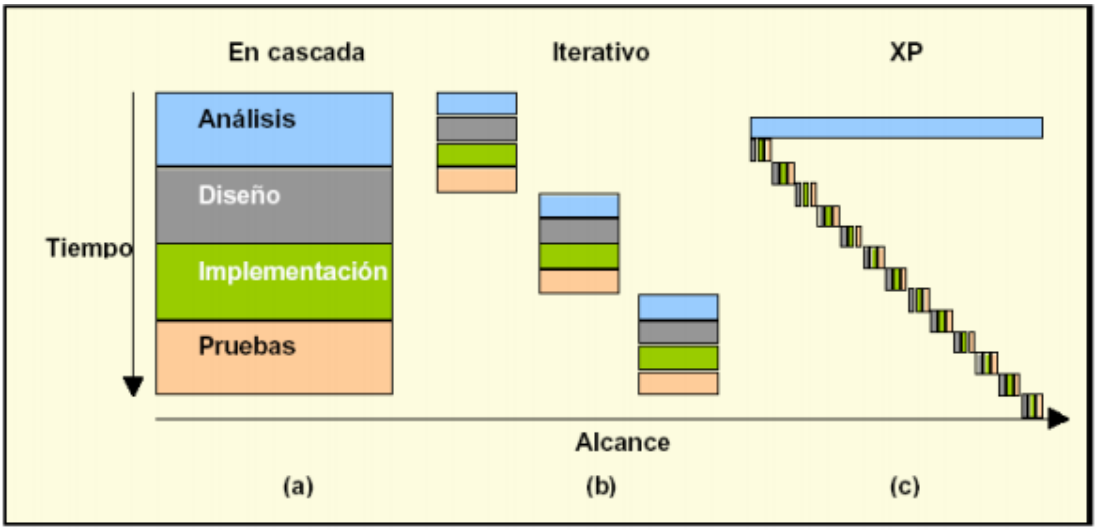


Ilustración 2-1: Comparación de fases entre otras metodologías y XP.

Fuente: (Joskowicz, 2008) Modelo XP

2.1.2.1. Fase de Exploración

Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de sencillas "historias de usuarios". Los programadores estiman los tiempos de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que estarán basadas en datos de muy alto nivel), y podrían variar cuando se analicen más en detalle en cada iteración. Esta fase dura típicamente un par de semanas, y el resultado es una visión general del sistema, y un plazo total estimado. (Joskowicz, 2008)

2.1.2.2. Fase de Planificación

La planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y, asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación.

Las “historias de usuarios” sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios. (Joskowicz, 2008)

2.1.2.3. Fase de Iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo. Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance. (Joskowicz, 2008)

2.1.2.4. Fase de Puesta en Producción

Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa. En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste (“fine tuning”). (Joskowicz, 2008)

2.2. Software Libre

El término “Software Libre” se refiere al software que respeta la libertad de los usuarios y de la comunidad. De manera general, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De esta manera, “software libre” se trata de libertad, no de precio. (GNU Operating System, 2017)

2.2.1. Las cuatro libertades esenciales

Un programa es software libre si los usuarios del programa tienen las cuatro libertades esenciales:

- Libertad para ejecutar el programa como lo desee, para cualquier propósito (libertad 0).
- La libertad de estudiar como el programa trabaja, y cambiarlo a conveniencia para que haga lo que los usuarios deseen (libertad 1). Acceso al código fuente es una precondición para esto.
- La libertad de redistribuir copias (libertad 2)
- La libertad de distribuir copias a otros usuarios con versiones modificadas (libertad 3). Haciendo esto se puede dar, a la comunidad entera, una oportunidad de beneficiarse de los cambios realizados. Acceso al código fuente es una precondición para esto.

(GNU Operating System, 2017)

2.3. XAMPP

XAMPP es un paquete de instalación independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web y los intérpretes para lenguajes de script: PHP y Perl. El programa se distribuye bajo la licencia GNU y actúa

como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. (Wikipedia, 2017)

Para su instalación solo se requiere su descarga e iniciar el asistente, el mismo que guía al usuario paso a paso, seleccionando los componentes que se desea instalar y el directorio de instalación.

XAMPP es una compilación de software libre (similar a una distribución de Linux). Es gratuita y puede ser copiada libremente de acuerdo a la licencia GNU GPL. Únicamente la compilación de XAMPP está publicada bajo la licencia GPL. Cada uno de los componentes incluidos tiene su propia licencia y el interesado debería consultarlas para conocer qué es posible y que no. En el caso de uso comercial se deberá consultar las licencias individuales, en particular MySQL. Desde el punto de vista de XAMPP como compilación, el uso comercial es gratuito. (Apache Friends, 2017)

2.4. Lenguage HTML

2.4.1. Definición

HTML es el lenguaje de marcado estándar usado para crear páginas web y sus elementos forman los cimientos de todos los sitios web. Es el lenguaje de la web para la creación de contenido para que todos puedan usar en cualquier lugar. (World Wide Web Consortium, 2017)

HTML es un lenguaje de marcas (etiquetas) que se emplea para dar formato a los documentos que se quieren publicar en la WWW. Los navegadores pueden interpretar las etiquetas y muestran los documentos con el formato deseado. (Luján Mora, 2001)

2.4.2. Historia Del HTML

La primera versión de HTML fue descrita por Tim Berners-Lee en 1991. En los primeros 5 años (1990 – 1995), HTML pasó por varias revisiones y

experimentó un número de extensiones, principalmente conducidas por el CERN, y luego por el IETF. (World Wide Web Consortium, 2015)

Con la creación de la W3C, el desarrollo del HTML se estandarizó, pero también sufrió de varios cambios. La siguiente tabla, muestra las versiones con sus respectivos años de lanzamiento:

VERSION	AÑO DE PUBLICACIÓN
HTML +	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999

Tabla 2-1: Versiones vs. Años de publicación del HTML (World Wide Web Consortium, 2015)

Fuente: (World Wide Web Consortium, 2015)

2.4.3. Etiquetas HTML

Los elementos en HTML son definidos por etiquetas, y son componentes vitales de este lenguaje. Constan de dos propiedades básicas: los atributos y el contenido.

Generalmente, cada elemento tiene una etiqueta de inicio y una de cierre, con sus *atributos* especificados dentro de la etiqueta de inicio, y el *contenido* ubicado entre las etiquetas de inicio y cierre (p. ej. Saludos amigos!) (Wikipedia, 2015).

Entre las etiquetas más usadas en HTML se puede encontrar , <br>, <title>, <body>, <b>, <form>, <input>, <textarea>, entre muchas otras (ver Anexo 5 – Manuales y tablas de referencia).

2.4.4. HTML 5

HTML5 es la última versión de HTML. El término representa dos conceptos diferentes:

- Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos.
- Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama HTML5 y amigos, a menudo reducido a HTML5.

(Mozilla Developer Network, 2017)

HTML5 no se vincula a una sintaxis o a otra, sino que admite ambas (hablando de html y xhtml).

Actualmente los navegadores corrigen los errores de sintaxis de distinta manera, y HTML5 trata de poner fin a esa necesidad de ingeniería inversa de los navegadores, que compiten por definir como se deben subsanar los errores.

(Franganillo, 2010)

A continuación, se listan algunos recursos sobre las tecnologías de HTML5, clasificados en varios grupos según su función:

- Semántica: Permite describir con mayor precisión cuál es su contenido.
- Conectividad: Permite comunicarse con el servidor de formas nuevas e innovadoras.
- Sin conexión y almacenamiento: Permite a las páginas web almacenar datos localmente en el lado del cliente y operar sin

conexión de manera más eficiente.

- Multimedia: Otorga un excelente soporte para utilizar contenido multimedia como lo son audio y video nativamente.
- Gráficos y efectos 2D/3D: Proporciona una amplia gama de nuevas características que se ocupan de los gráficos en la web como lo son canvas 2D, WebGL, SVG, etc.
- Rendimiento e Integración: Proporciona una mayor optimización de la velocidad y un mejor uso del hardware.
- Acceso al dispositivo: Proporciona APIs para el uso de varios compomentes internos de entrada y salida de nuestro dispositivo.
- CSS3: Ofrece una nueva gran variedad de opciones para hacer diseños más sofisticados.

(Mozilla Developer Network, 2017)

2.5. CSS

Hojas de Estilo en Cascada, es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne.

(World Wide Web Consortium - Oficina española, 2017)

2.6. PHP

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser incrustado en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender (PHP, 2015).

PHP facilita el desarrollo web, haciendo que las páginas desarrolladas en este lenguaje se pueda incluir HTML. El código PHP se escribe entre etiquetas especiales, como se ve en el siguiente ejemplo:

```
<html>
   <head>
      <title>Ejemplo Código HTML y PHP</title>
   </head>
   <body>

      <?php

         echo "Pontificia Universidad Católica del Ecuador";

      ?>

   </body>
</html>
```

Ilustración 2-2: Ejemplo de codificación en el lenguaje PHP.

Fuente: Autor de la disertación

Es un lenguaje del lado del servidor[3], es decir que su código es interpretado por un servidor web con un módulo de procesador PHP instalado, generando la página web que es desplegada en el navegador, sin que el cliente pueda determinar que código ha producido dicho resultado.

Además del uso para escribir scripts del lado de servidor, PHP posee una línea de comandos con la cual también se puede escribir scripts. Para ello se debe contar únicamente con el intérprete PHP para usarlo de esta manera. En la documentación oficial de PHP existe toda una sección dedicada al procesamiento desde la línea de comandos.

Otro de los usos que se le puede dar a PHP es la escritura de aplicaciones de interfaz gráfica con la extensión PHP-GTK. Esta extensión no es parte de la distribución principal.

Una de las ventajas de este lenguaje es que puede ser usado en la mayoría de los servidores web, al igual que en casi todos los sistemas operativos y plataformas sin ningún costo, por ejemplo Linux, Unix, Microsoft Windows, MAC OS X, RISC OS, etc. al igual que en varios servidores web como Apache, IIS, etc.

2.6.1. Historia

Rasmus Lerdorf, programador canadiense, fue el creador de este lenguaje. Desarrollado originalmente en Perl, PHP se usó en sus inicios para mostrar la hoja de vida de Rasmus, y guardar dinámicamente datos como el tráfico de su sitio web.

[3] Tecnología que consiste en el procesamiento de una petición de un usuario mediante la interpretación de un script en el servidor web para generar páginas HTML dinámicamente como respuesta.

El 8 de junio de 1995 se publicó el código fuente de “Personal Home Page Tools”, y combinado con su propio Form Interpreter se crea PHP/FI, originalmente escrito en Lenguaje C.

En 1997 se reescribe el parser (analizador sintáctico), creando la base de la 3era versión de PHP, cambiando el nombre a PHP Hypertext Preprocessor, el cual fue publicado con una versión oficial en junio de 1998.

En 1999, se reescribe el código de PHP produciendo *motor Zend* y se funda *Zend Technologies* en Ramat Gan, Israel.

En mayo del 2000 se lanza PHP 4; y 7 años después, Zend anuncia la suspensión del soporte y desarrollo para esa versión.

El 13 de julio del 2004, PHP 5 sale a la luz, con significativas mejoras como soporte para la programación orientada a objetos, rendimiento, soporte para MySQL con la extensión completamente reescrita, soporte para XML, soporte nativo para SQLite, soporte integrado para SOAP, manejo de excepciones, mejoras en la implementación con Oracle, entre otras.

(PHP, 2015)

2.7. MySQL

MySQL, el sistema de gestión de base de datos de SQL de código abierto más popular, es desarrollado, distribuido y soportado por Oracle Corporation. (Oracle Corporation, 2017)

2.7.1. Características

Las principales características de MySQL son:

- Es un sistema de gestión de base de datos.
- Las bases de datos de MySQL son relacionales.
- Es de código abierto.

- El servidor de base de datos MySQL es rápido, confiable, escalable y de fácil uso.
- Funciona en sistemas cliente/servidor o embebidos.
- Una gran cantidad de software de contribución de MySQL está disponible

(Oracle Corporation, 2017)

2.7.2. Licencia

El software MySQL es licencia doble. Los usuarios pueden elegir usar el software MySQL como un producto Open Source bajo los términos de la Licencia Pública General de GNU o comprar su versión comercial. (Oracle Corporation, 2017)

2.8. AJAX y Javascript

Javascript es un lenguaje de scripting que se ejecuta del lado del cliente, y su función principal es acceder a los elementos del DOM para brindar al usuario una mayor interactividad con la página web. Está basado en prototipos, es orientado a objetos y es dinámico e imperativo, es decir que le indican al computador como realizar una tarea.

Actualmente los navegadores más usados[4] pueden interpretar este lenguaje. Sin embargo, en sus inicios, Javascript era menospreciado por los desarrolladores ya que este lenguaje era dirigido para publicadores de artículos y otros aficionados (Crockford, 2015), pero este rechazo cambió con la llegada de AJAX.

[4] Navegadores más usados según (StatCounter, 2017): Chrome, Firefox, Internet Explorer, Safari, Opera

AJAX, por sus siglas en inglés de Asynchronous Javascript And XML (Javacript asíncrónico y XML), es una técnica de desarrollo para crear aplicaciones que se ejecutan en segundo plano del lado del cliente a través del navegador web. Con esto se logra una mejor experiencia del usuario ya que mejora velocidad e interactividad en la página web que se use.

Combina 5 tecnologías: HTML, CSS, DOM, XML HttpRequest y XML. Las dos primeras se pueden combinar en una sola llamada XHTML, y son usadas para la presentación de la data. El DOM interactúa con la información desplegada mediante un lenguaje de scripting como Javascript. El objeto XML HttpRequest enlaza el cliente y el servidor web mediante una comunicación asíncrona. Normalmente el formato de la data enviada en esta comunicación es XML, pero otros formatos como JSON, texto plano, etc. son válidos.

2.8.1. Uso de AJAX en la programación web

Un navegador web es el entorno de acogida más común para JavaScript. (Wikipedia, 2015)

El uso de AJAX y Javascript en páginas HTML se centra más en la experiencia del usuario en cuanto a interactividad se refiere:

Por ejemplo, en la red social Facebook, al dar un “Like” a un comentario, se usa AJAX para no recargar la página. El mismo concepto se usa para la actualización del estado, el envío de comentarios, la actualización de las publicaciones en el muro, entre otras acciones de la misma red.

Javascript también puede crear animaciones para los objetos una página web, cambiando su color y tamaño, escondiéndolos y cambiándolos de lugar. Se puede combinar estas características en la creación de juegos interactivos incluyendo la reproducción de audio y video.

La validación de formularios es uno de los usos más comunes de javascript. Esto se refiere al análisis de los datos ingresados por un usuario en un formulario web antes de ser enviados al servidor.

2.9. CodeIgniter: Framework de desarrollo

2.9.1. Definición del término “framework”

Un framework o entorno de trabajo es un conjunto de módulos y librerías de funciones y/o clases desarrolladas que sirven de base para el desarrollo de software, es decir, que es un programa con el que se puede construir otros programas.

2.9.2. CodeIgniter

Aplicación web de código abierto desarrollada en el lenguaje de programación PHP, que sirve de base para la creación de nuevas aplicaciones en PHP basándose en el estándar de programación MVC (modelo – vista - controlador) y proponiendo una estructura para la organización de los scripts.

2.9.3. Características generales

Su instalación y configuración permiten a los desarrolladores crear su entorno de trabajo en pocos minutos, y la ligereza del núcleo este framework permite una rápida ejecución de la aplicación desarrollada. Además, está bien documentado, lo que facilita la adopción de este framework para el desarrollo de aplicaciones web.

2.9.4. Breve historia

Code Igniter nace en el año 2006 a mano de Rick Ellis. Esencialmente es una colección de clases refactorizadas escritas originalmente para el CMS de Ellislab. Fue creado para ser un toolkit simple y elegante, que permitiese el desarrollo de aplicaciones y sitios web, atrayendo a miles de desarrolladores PHP.

En el 2008 se convirtió en el líder de la industria de los frameworks, ya que era uno de los pocos que estaba bien documentado, recibiendo una felicitación por parte de Rasmus Lerdorf, el creador de PHP.

En el 2009, Ellislab lanza ExpressionEngine 2.0 trabajado totalmente en CodeIgniter. Este flexible código base permitía a desarrolladores externos a construir herramientas y add-ons para ExpressionEngine, dando como resultado una explosión de talento en la comunidad de ExpressionEngine.

Desde el 2014, Ellislab otorgó CodeIgniter al *British Columbia Institute of Technology* quienes continuaron con el desarrollo de esta herramienta convirtiéndola en un laboratorio viviente.

(EllisLab, 2015)

2.9.5. Pasos para la Instalación

Para la instalación de CodeIgniter existen 2 requisitos que debe cumplir el servidor:

- Tener instalada la versión de PHP 5.4 o superior
- Una base de datos es requerida para la mayoría de las aplicaciones web. Para este proyecto se usa MySQL.

Una vez que se cuente con un servidor que cumpla con los requisitos anteriormente mencionados, el siguiente paso es descargar el paquete desde el sitio web, y seguir los siguientes pasos:

1. Descomprimir el paquete descargado.

2. Cargar los archivos y carpetas de CodeIgniter al servidor. Normalmente el archivo index.php estará en el directorio raíz.

3. Con un editor de texto, abrir el archivo application/config/config.php y configurar la URL base. En el caso de necesitar sesiones encriptadas, se debe configurar también la clave de encriptación en el mismo archivo.

4. En el caso de que ser requiera usar una base de datos, con un editor de texto abrir el archivo application/config/database.php y escribir la configuración de la base de datos.

2.9.6. Flujo de la Aplicación

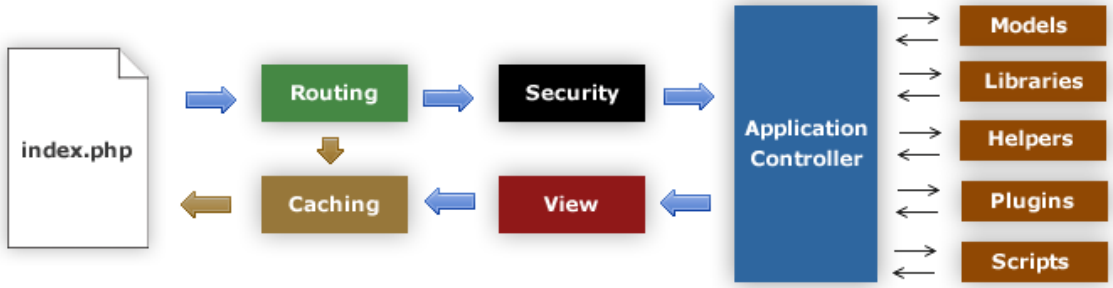


Ilustración 2-3: Flujo de una Aplicación en CodeIgniter

Fuente: (CodeIgniter, 2015) Application FlowChart

- El archivo index.php sirve de controlador frontal, inicializando la base de recursos necesarios para ejecutar CodeIgniter.
- El router examina la petición HTTP para determinar que debería hacerse con esta.
- Si el archivo de caché existe, este es enviado al navegador. Funciona como bypass, saltándose toda la ejecución normal del sistema.
- Seguridad. Antes de que el controlador de la aplicación sea cargado, la petición HTTP y cualquier información enviada es filtrada por seguridad.
- El controlador carga el modelo, las librerías core, helpers y cualquier otro recurso necesario para procesar esa petición.
- La vista final es renderizada y enviada al navegador web para ser vista. Si la caché está habilitada, la vista es primeramente cacheada para las peticiones subsecuentes.

(CodeIgniter, 2015, Aplication FlowChart)

2.9.7. MVC (Modelo-Vista-Controlador)

CodeIgniter está basado en el patrón de desarrollo Modelo – Vista - Controlador. MVC es un enfoque de software que separa la lógica de la aplicación, de la presentación de la misma (…).

Modelo: Se refiere a la estructura que tengan los datos.

Vista: Es la presentación de la aplicación al usuario. Gestiona como se muestran los datos.

Controlador: sirve como intermediario entre la vista y el modelo, cargando todos los recursos necesarios para procesar la petición HTTP.

Como ventajas del uso de este patrón de diseño se puede listar las siguientes:

- Diferentes vistas para un modelo.
- Nuevas vistas sin necesidad de modificar el modelo.
- Configuración para componentes complejos mucho más manejable.
- Seguridad.

(CodeIgniter, 2015)

CAPÍTULO 3. ANÁLISIS DE REQUERIMIENTOS

En este capítulo se describen los requerimientos funcionales y no funcionales para el desarrollo del software, explicando las técnicas que se usaron para levantar dichos requerimientos.

3.1. Definiciones

El levantamiento de requerimientos es uno de los procesos más importantes, pero a su vez más críticos, dentro del desarrollo de software y que requiere mayor conocimiento ya que es donde se define el diseño de la solución según las necesidades del cliente. El no entender bien un requerimiento conlleva al gasto extra de recursos. (Martínez Guerrero & Silva Delgado, 2011)

En *extreme programming*, para la ingeniería de requerimientos se usan las historias de usuario. Una historia de usuario es la descripción de un requisito del sistema, escrito en pocas líneas, con lenguaje común en el entorno del cliente. Deben ser escritas, o en el peor de los casos, validadas por el cliente. Una historia de usuario describe la funcionalidad deseada desde la perspectiva del cliente. (Hanan Tek, 2010)

Historia de Usuario	
Número:[Id.]	**Nombre Historia de Usuario:**
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario:[Quien la genera o usa]	**Iteración Asignada:**
Prioridad en Negocio: **(Alta / Media / Baja)**	**Puntos Estimados:[Semanas de desarrollo, un punto es una semana efectiva de desarrollo]**
Riesgo en Desarrollo: **(Alto / Medio / Bajo)**	**Puntos Reales:**
Descripción:[Descripción de la tarea o proceso que da lugar a esta historia de usuario]	
Observaciones:	

Ilustración 3-1: Estructura básica de una historia de usuario

Fuente: (Hanan Tek, 2010)

3.2. Requerimientos no funcionales

Como condicionantes para el desarrollo del software se tienen los siguientes puntos a considerar:

- El entorno de funcionamiento del software debe ser web, ya que los socios deben poder consultar el estado de su red en cualquier momento y desde cualquier lugar.
- El desarrollo del prototipo debe desarrollarse con herramientas que no requieran la compra de licencias ya que generaría un costo adicional para el cliente y para el desarrollador.

3.3. Requerimientos Funcionales

Se realizaron 7 entrevistas en total, de las cuales 6 fueron personas involucradas en las redes de mercadeo, y 1 persona que deseaba implementar el sistema de compensación uninivel simple con compactación en su negocio. Además de pedirles que aprueben las historias de usuario generadas como parte del levantamiento de requerimientos de la

metodología *extreme programming*, a todos se les hizo el siguiente cuestionario:

1.- Si es que es de su conocimiento, explique cómo funciona el sistema de compensación uninivel con compactación de socios.

2.- ¿Para qué negocios cree que sería factible una implementación de este sistema, y en qué circunstancias?

3.- ¿En un negocio que se implemente este tipo de plan de compensación, quienes cree que serían los actores principales/secundarios y por qué?

4.- ¿Cuál sería el porcentaje de pago para los socios/patrocinados?

5.- Acotaciones que pueda aportar referente al tema.

3.3.1. Resumen de las entrevistas

Las conclusiones para cada una de las 5 preguntas son las siguientes:

Funcionamiento del sistema uninivel simple con compactación de socios

El sistema de compensación uninivel simple con compactación se lo puede explicar mejor con el ejemplo de un gimnasio que está implementando este sistema, y como compensación entrega $5 mensuales a cada socio que traiga un socio nuevo:

- Cualquier persona puede ser socio del gimnasio, es decir, inscribirse al gimnasio.

- Cualquier socio puede acceder a este sistema de compensación.

- Si un socio A invita a una persona B a ser socio del gimnasio, y la persona B se inscribe, entonces B es el “recomendado” de A, y A gana $5 por haber ingresado un socio nuevo al gimnasio.

- El socio A puede invitar a cualquier cantidad de personas a ser socias del gimnasio, y va a ganar $5 mensuales por cada persona (de las que ha invitado) que se convierta en socio. Todas las nuevas personas invitadas por A estarán “abajo” de A, es decir, en su primer nivel.

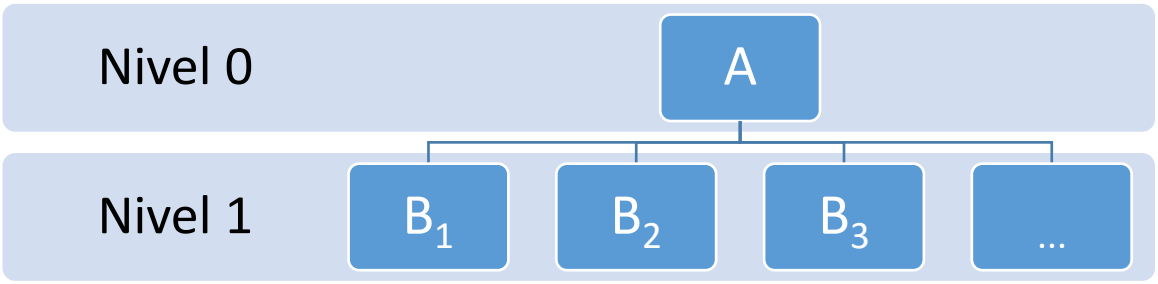


Ilustración 3-2: Sistema de compensación uninivel con compactación simple

Fuente: Autor del proyecto de disertación de grado

- Si el nuevo socio B (quien está debajo de A) invita a otras personas, también ganará $5 por cada persona que se convierta en socio. Sin embargo, A no ganará nada por las personas que B invite, ya que esas personas estarán en su 2do nivel. Este esquema de compensación es llamado uninivel simple.

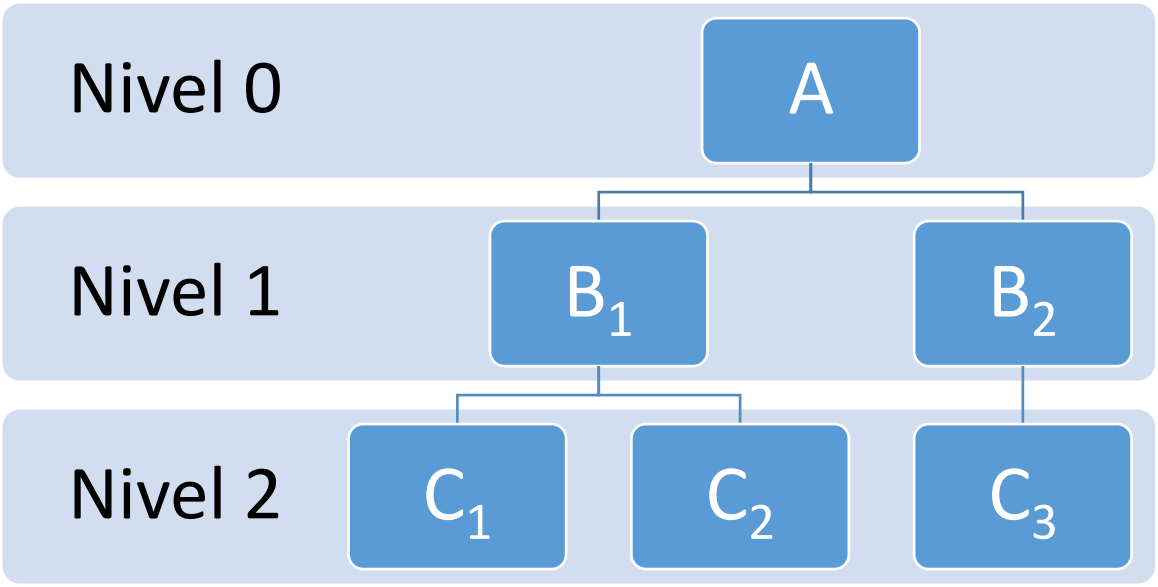


Ilustración 3-3: Sistema de compensación uninivel con compactación simple – Jerarquía

Fuente: Autor del proyecto de disertación de grado

- Si un socio se retira, o queda inactivo, todos los socios que estaban en su primer nivel pasan a formar parte del socio que está en el nivel inmediato superior. Así en el ejemplo, si B1 se retira del gimnasio, C1 y C2 pasan a formar parte del Nivel 1 de A, haciendo que A gane $5 por cada socio (total $10), cumpliendo con la COMPACTACIÓN.

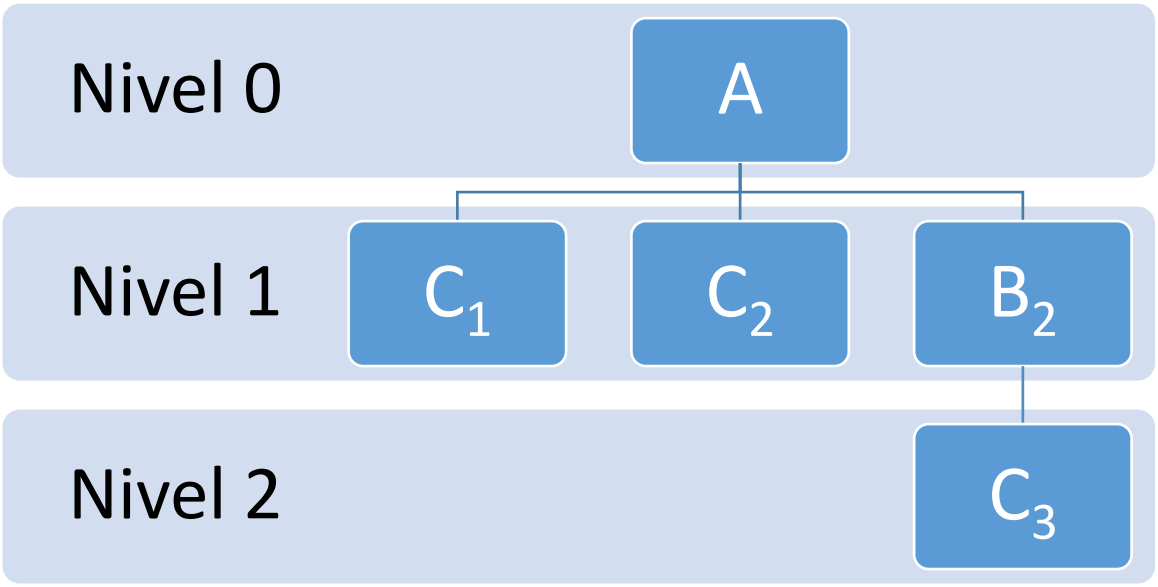


Ilustración 3-4: Sistema de compensación uninivel con compactación simple - Compactación

Fuente: Autor del proyecto de disertación de grado

Tipos de negocios para implementar el sistema

Los negocios para los que recomiendan los entrevistados la implementación de este sistema son negocios que tengan suscripciones periódicas o recursivas (p. ej. mensuales) a sus productos o servicios, por ejemplo:

- Gimnasios, clubes deportivos
- Academias de baile, artes marciales
- Academias de educación / capacitación continua
- Revistas, diarios
- Telefonía móvil, televisión por cable, internet
- Seguros de salud, de vida, de autos, etc.
- Clubes sociales

Actores principales y secundarios

Como actores principales se tienen a todos los socios, y al administrador del negocio por razones obvias. Como actores secundarios: dueño del negocio, los prospectos, que son los posibles nuevos socios, interesados en el producto o servicio, ya que se puede armar una base de datos y tomar acciones posteriores para que se conviertan en socios. Finalmente, los empleados del negocio, ya que pueden formar parte del sistema como promotores.

Límite del porcentaje de pago

El porcentaje de pago para este tipo de sistemas “no tiene límite”, pero siempre va a depender de los costos del producto o servicio.

Otros puntos relevantes a tomar en cuenta

El sistema no debe ser mezclado con otro sistema de compensación sin antes haber analizado las consecuencias económicas y legales de la generación de un plan híbrido con un experto en el diseño de sistemas de compensación para redes de mercadeo.

La implementación de este tipo de sistemas (MLM) debe ser bastante exhaustiva en la capacitación a los dueños y administradores del negocio, y una buena comunicación/difusión del funcionamiento correcto de este sistema.

Es recomendable que se cuente con un aval. En Ecuador existe la AEVD, la cual avala la legalidad y buen funcionamiento de cualquier negocio que implemente este tipo de sistemas y que formen parte de esta asociación.

3.3.2. Historias de usuarios

Historia de usuario No. 1 - Gestionar socios

Historia de Usuario	
Número: 1	**Nombre:** Gestionar Socios
Usuario: Administrador	**Iteración asignada:** 1
Prioridad: Alta	**Puntos estimados:** 1
Riesgo: Medio	**Puntos reales:** 1
Descripción: Como administrador del sistema Quiero poder crear, modificar, eliminar y listar socios del sistema Para generar la red del negocio, y dar o quitar acceso a los socios al sistema	
Desarrollador: Douglas Jara Negrete	

Tabla 3-1: Historia de usuario No. 1 – Gestionar socios.
Fuente: Autor del proyecto de disertación.

Historia de usuario No. 2 - Gestionar red de socios

Historia de Usuario	
Número: 2	**Nombre:** Gestionar red de socios
Usuario: Administrador	**Iteración asignada:** 1
Prioridad: Alta	**Puntos estimados:** 1
Riesgo: Medio	**Puntos reales:** 1
Descripción: Como administrador del sistema Quiero poder modificar la red de socios Para mantener el plan de compensación como parte del modelo de negocio	
Desarrollador: Douglas Jara Negrete	

Tabla 3-2: Historia de usuario No. 2 – Gestionar red de socios.
Fuente: Autor del proyecto de disertación.

Historia de usuario No. 3 - Obtener red de un socio

Historia de Usuario	
Número: 3	**Nombre:** Obtener red de un socio
Usuario: Socio	**Iteración asignada:** 1
Prioridad: Alta	**Puntos estimados:** 1
Riesgo: Medio	**Puntos reales:** 1
Descripción: Como usuario socio Quiero poder revisar mi red de socios Para estar informado de la red que he generado	
Desarrollador: Douglas Jara Negrete	

Tabla 3-3: Historia de usuario No. 3 – Obtener red de un socio.
Fuente: Autor del proyecto de disertación.

Historia de usuario No. 4 - Obtener comisiones de un socio

Historia de Usuario	
Número: 4	**Nombre:** Obtener comisiones de un socio
Usuario: Socio	**Iteración asignada:** 1
Prioridad: Alta	**Puntos estimados:** 1
Riesgo: Medio	**Puntos reales:** 1
Descripción: Como usuario socio Quiero poder revisar mis comisiones Para estar informado de las comisiones que he generado	
Desarrollador: Douglas Jara Negrete	

Tabla 3-4: Historia de usuario No. 4 – Obtener comisiones de un socio.
Fuente: Autor del proyecto de disertación.

Historia de usuario No. 5 – Obtener comprobante de pago

Historia de Usuario	
Número: 5	**Nombre:** Obtener comprobante de pago
Usuario: Facturación	**Iteración asignada:** 2
Prioridad: Alta	**Puntos estimados:** 1
Riesgo: Medio	**Puntos reales:** 1
Descripción: Como usuario de facturación Quiero poder generar una comprobante de pago Para generar comisiones de socios e imprimir comprobantes de pago	
Desarrollador: Douglas Jara Negrete	

Tabla 3-5: Historia de usuario No. 5 – Obtener comprobante de pago.
Fuente: Autor del proyecto de disertación.

Historia de usuario No. 6 – Gestionar ítems

Historia de Usuario	
Número: 6	**Nombre:** Gestionar ítems
Usuario: Administrador	**Iteración asignada:** 2
Prioridad: Alta	**Puntos estimados:** 1
Riesgo: Bajo	**Puntos reales:** 1
Descripción: Como administrador del sistema Quiero poder crear, modificar, eliminar y listar ítems del sistema Para que esos ítems puedan añadirse a un comprobante de pago	
Desarrollador: Douglas Jara Negrete	

Tabla 3-6: Historia de usuario No. 6 – Gestionar ítems.
Fuente: Autor del proyecto de disertación.

Historia de usuario No. 7 – Gestionar formas de pago

Historia de Usuario	
Número: 7	**Nombre:** Gestionar formas de pago
Usuario: Administrador	**Iteración asignada:** 2
Prioridad: Alta	**Puntos estimados:** 1
Riesgo: Bajo	**Puntos reales:** 1
Descripción: Como administrador del sistema Quiero poder crear, modificar, eliminar y listar formas de pago Para que se escojan el momento de generar un comprobante de pago	
Desarrollador: Douglas Jara Negrete	

Tabla 3-7: Historia de usuario No. 7 – Gestionar formas de pago.
Fuente: Autor del proyecto de disertación.

Historia de usuario No. 8 – Gestionar países

Historia de Usuario	
Número: 8	**Nombre:** Gestionar países
Usuario: Administrador	**Iteración asignada:** 3
Prioridad: Alta	**Puntos estimados:** 1
Riesgo: Bajo	**Puntos reales:** 1
Descripción: Como administrador del sistema Quiero poder crear, modificar, eliminar y listar países Para que se escojan el momento de gestionar un socio	
Desarrollador: Douglas Jara Negrete	

Tabla 3-8: Historia de usuario No. 8 – Gestionar países
Fuente: Autor del proyecto de disertación.

Historia de usuario No. 9 – Gestionar configuraciones del sistema

Historia de Usuario	
Número: 7	**Nombre:** Gestionar configuraciones del sistema
Usuario: Administrador	**Iteración asignada:** 3
Prioridad: Alta	**Puntos estimados:** 1
Riesgo: Bajo	**Puntos reales:** 1
Descripción: Como administrador del sistema Quiero poder crear, modificar, eliminar y listar configuraciones del sistema Para que sean fácilmente modificables por el administrador del sistema.	
Desarrollador: Douglas Jara Negrete	

Tabla 3-9: Historia de usuario No. 9 – Gestionar configuraciones del sistema.
Fuente: Autor del proyecto de disertación.

Historia de usuario No. 10 – Obtener estadísticas

Historia de Usuario	
Número: 7	**Nombre:** Obtener estadísticas
Usuario: Gerencia	**Iteración asignada:** 4
Prioridad: Alta	**Puntos estimados:** 1
Riesgo: Medio	**Puntos reales:** 1
Descripción: Como usuario de gerencia Quiero poder visualizar estadísticas básicas como # de socios, # de comisiones por pagar, total ventas Para estar informado del movimiento del negocio	
Desarrollador: Douglas Jara Negrete	

Tabla 3-10: Historia de usuario No. 10 – Obtener estadísticas.
Fuente: Autor del proyecto de disertación.

3.3.3. Resumen de requerimientos funcionales

Requerimiento	Descripción de la Interfase
Gestionar usuarios/socios	Creación, edición, eliminación y listado de usuarios/socios.
Gestionar la red de socios	Listar y modificar la red de cada socio.
Obtener comprobante de pago	Presentación de ítems para la obtención de un comprobante de pago y generación de comisiones para cada socio.
Gestionar ítems, formas de pago, países	Crear, modificar, eliminar y listar ítems, formas de pago y países.
Gestionar configuraciones del sistema	Crear, modificar, eliminar y listar configuraciones del sistema.
Obtener estadísticas	Presentar visualmente estadísticas básicas del negocio como número de socios, comisiones, total de ventas.

Tabla 3-11: Resumen de Requerimientos

Fuente: Autor del proyecto de disertación

CAPÍTULO 4. DISEÑO DEL PROTOTIPO

4.1. Diseño de la base de datos

4.1.1. Diseño conceptual

A continuación, se muestra el esquema conceptual del prototipo generado acorde a los requerimientos obtenidos:

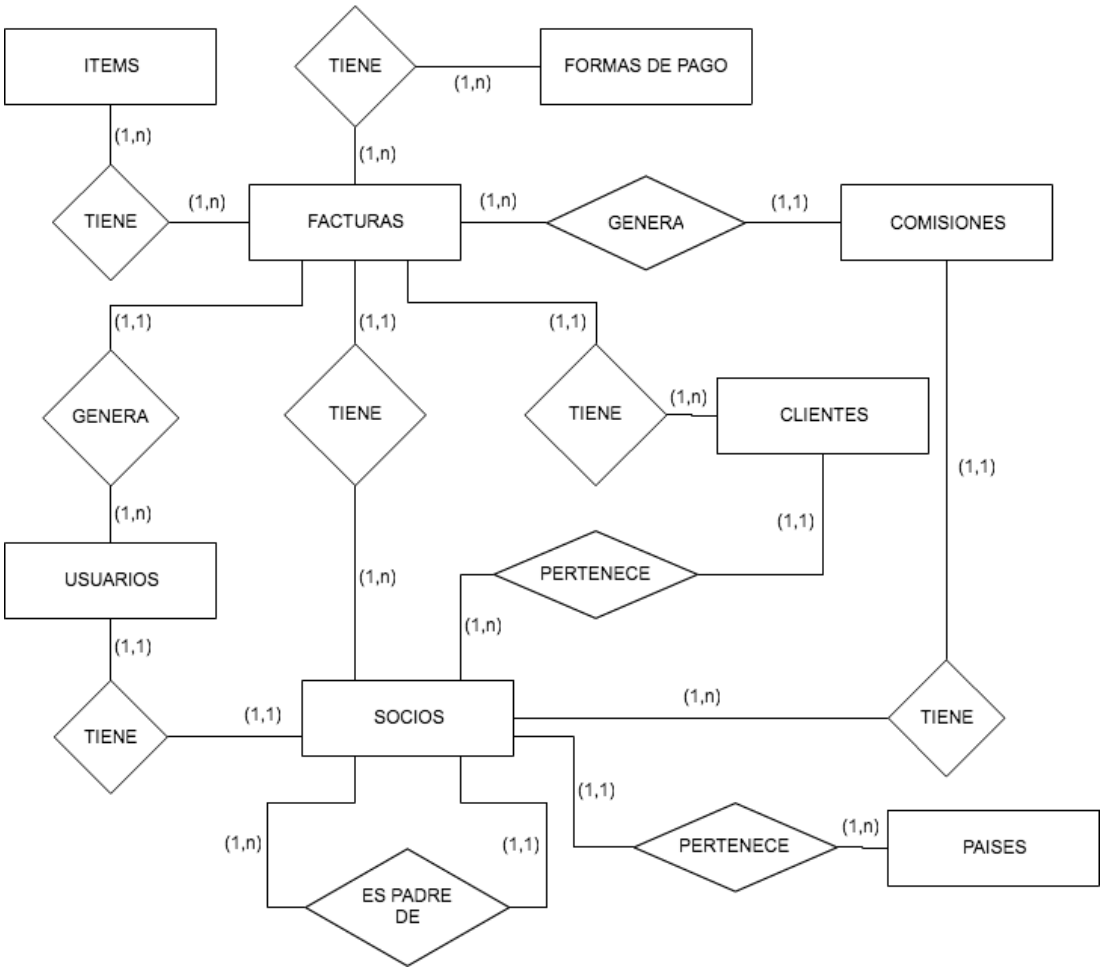


Ilustración 4-1: Esquema conceptual del prototipo

Fuente: Autor del proyecto de disertación

4.1.2. Diseño lógico

El diseño define un total de 15 tablas, que cumple con la 3era forma normal (3FN).

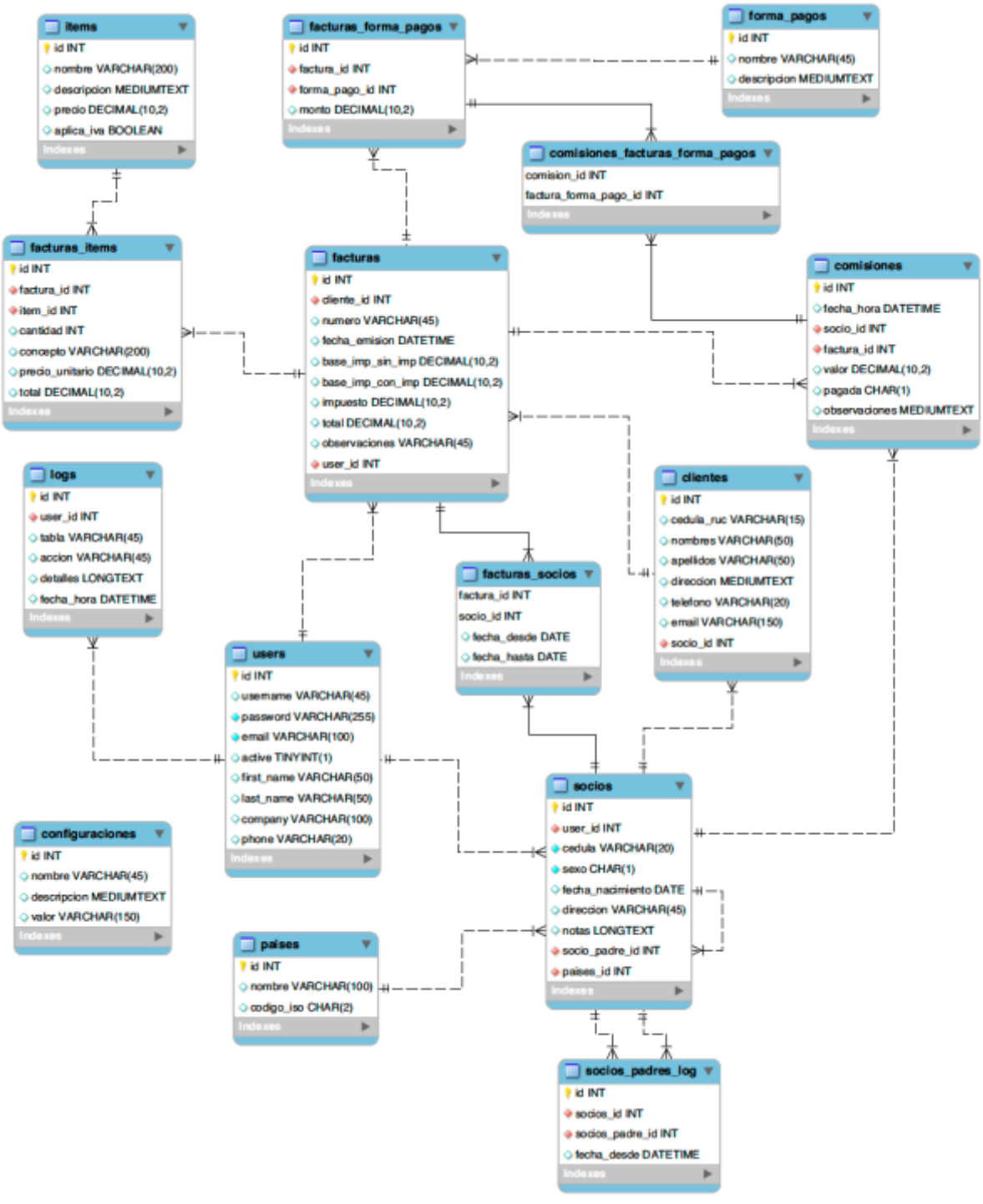


Ilustración 4-2: Diseño lógico del prototipo

Fuente: Autor del proyecto de disertación

4.1.3. Diseño físico

A continuación, se describe el diccionario de datos. El script SQL para generar la base de datos se encuentra como anexo a esta disertación.

Tabla clientes

Campo	Tipo de dato	Descripción
id	int(11)	Identificador del cliente
cedula	varchar(15)	Cédula o RUC del cliente
nombres	varchar(50)	Nombres del cliente
apellidos	varchar(50)	Apellidos del cliente
direccion	mediumtext	Dirección del cliente
telefono	varchar(20)	Teléfono del cliente
email	varchar(150)	Dirección electrónica del cliente
socio_id	Int(11)	Identificador del socio al que hace referencia el cliente

Tabla 4-1: Tabla clientes

Fuente: Autor del proyecto de disertación

Tabla comisiones

Campo	Tipo de dato	Descripción
id	int(11)	Identificador de la comisión
fecha_hora	datetime	Fecha hora en la que se registró la comisión
socio_id	int(11)	Identificador del socio al que pertenece la comisión
factura_id	int(11)	Identificador de la factura que generó la comisión
valor	decimal(10,2)	Valor de la comisión
pagada	char(1)	Indica si la comisión ya ha sido pagada (1 si, 0 no)
observaciones	mediumtext	Observaciones del registro de de la comisión

Tabla 4-2: Tabla comisiones

Fuente: Autor del proyecto de disertación

Tabla comisiones_facturas_forma_pagos

Campo	Tipo de dato	Descripción
comision_id	int(11)	Identificador de la comisión
factura_forma_pago_id	int(11)	Identificador de la forma de pago de una factura

Tabla 4-3: Tabla comisiones_facturas_forma_pagos

Fuente: Autor del proyecto de disertación

Tabla configuraciones

Campo	Tipo de dato	Descripción
id	int(11)	Identificador de la comisión
nombre	varchar(45)	Nombre de la variable de configuración
descripcion	mediumtext	Descripción de la variable de configuración
valor	varchar(150)	Valor de la variable de configuración

Tabla 4-4: Tabla configuraciones

Fuente: Autor del proyecto de disertación

Tabla facturas

Campo	Tipo de dato	Descripción
id	int(11)	Identificador de la factura
cliente_id	int(11)	Identificador del cliente al que se le emíte la factura
numero	varchar(45)	Número de la factura
fecha_emision	datetime	Fecha y hora en la que se emite la factura
base_imp_sin_imp	decimal(10,2)	Base imponible sin impuestos
base_imp_con_imp	decimal(10,2)	Base imponible con impuestos
impuesto	decimal(10,2)	Valor del impuesto de la factura
total	decimal(10,2)	Total de la factura
observaciones	mediumtext	Observaciones de la factura
user_id	int(11)	Identificador del usuario que genera la

Campo	Tipo de dato	Descripción
		factura

Tabla 4-5: Tabla configuraciones

Fuente: Autor del proyecto de disertación

Tabla facturas_foma_pagos

Campo	Tipo de dato	Descripción
id	int(11)	Identificador de la tabla
factura_id	int(11)	Identificador de la factura
forma_pago_id	int(11)	Identificador de la forma de pago usada en la factura
monto	decimal(10,2)	Monto que el cliente pago con la forma de pago

Tabla 4-6: Tabla facturas_forma_pago

Fuente: Autor del proyecto de disertación

Tabla facturas_items

Campo	Tipo de dato	Descripción
id	int(11)	Identificador de la tabla
factura_id	int(11)	Identificador de la factura
item_id	int(11)	Identificador del ítem
cantidad	decimal(10,2)	Cantidad de ítems
concepto	varchar(200)	Concepto
precio_unitario	decimal(10,2)	Precio unitario del ítem
total	decimal(10,2)	Valor total del ítem (cantidad x precio_unitario)

Tabla 4-7: Tabla facturas_items

Fuente: Autor del proyecto de disertación

Tabla facturas_socios

Campo	Tipo de dato	Descripción
factura_id	int(11)	Identificador de la factura
socio_id	int(11)	Identificador del socio
fecha_desde	date	Fecha desde cuando es válida la factura para el socio
fecha_hasta	date	Fecha límite en la que es válida la factura para el socio

Tabla 4-8: Tabla facturas_socios

Fuente: Autor del proyecto de disertación

Tabla forma_pagos

Campo	Tipo de dato	Descripción
id	int(11)	Identificador de la forma de pago
nombre	varchar(45)	Nombre de la forma de pago
descripcion	mediumtext	Descripción de la forma de pago

Tabla 4-9: Tabla forma_pagos

Fuente: Autor del proyecto de disertación

Tabla items

Campo	Tipo de dato	Descripción
id	int(11)	Identificador del ítem
nombre	varchar(200)	Nombre del ítem
descripcion	mediumtext	Descripción del ítem
precio	decimal(10,2)	Precio del ítem
aplica_iva	titnyint(1)	Indica si al ítem se le aplica el IVA (1 si, 0 no)

Tabla 4-10: Tabla Ítems

Fuente: Autor del proyecto de disertación

Tabla paises

Campo	Tipo de dato	Descripción
id	int(11)	Identificador del país
nombre	varchar(100)	Nombre del país
codigo_iso	char(2)	Código ISO 3166-1 alfa 2 del país

Tabla 4-11: Tabla paises

Fuente: Autor del proyecto de disertación

Tabla socios

Campo	Tipo de dato	Descripción
id	int(11)	Identificador del socio
user_id	int(11)	Identificador del usuario del socio
cedula	varchar(15)	Cédula o RUC del socio
sexo	char(1)	Sexo del socios (H hombre, M mujer)
fecha_nacimiento	date	Fecha de nacimiento del socio
direccion	longtext	Dirección del socio
pais_id	int(11)	Identificador del país del socio
notas	longtext	Notas u observaciones
socio_padre_id	int(11)	Identificador del socio padre (o patrocinador) del socio
fecha_ingreso	date	Fecha en la que ingresó el socio

Tabla 4-12: Tabla socios

Fuente: Autor del proyecto de disertación

Tabla socios_padres_log

Campo	Tipo de dato	Descripción
id	int(11)	Identificador del log
socio_id	int(11)	Identificador del socio
socio_padre_id	varchar(15)	Identificador del socio padre
fecha_desde	char(1)	Fecha desde cuando socio_padre_id es padre de socio_id

Tabla 4-13: Tabla socios_padres_log

Fuente: Autor del proyecto de disertación

Tabla users

Campo	Tipo de dato	Descripción
id	int(11)	Identificador del usuario
username	varchar(100)	Nombre de usuario del usuario
password	varchar(255)	Clave de acceso del usuario
email	varchar(100)	Email del usuario
active	tinyint(1)	Indicador para revisar si el usuario está activo (1 si, 0 no)
first_name	varchar(50)	Nombres del usuario
last_name	varchar(50)	Apellidos del usuario
company	varchar(100)	Empresa del usuario
phone	varchar(20)	Teléfono del usuario

Tabla 4-14: Tabla users

Fuente: Parte de la Estructura del módulo IonAuth para CodeIgniter

Tabla users_groups

Campo	Tipo de dato	Descripción
id	int(11)	Identificador de la tabla
user_id	int(11)	Identificador del usuario
group_id	mediumint(8)	Identificador del grupo al que pertenece el usuario

Tabla 4-15: Tabla users_groups

Fuente: Parte de la Estructura del módulo IonAuth para CodeIgniter

Tabla groups

Campo	Tipo de dato	Descripción
id	int(11)	Identificador del grupo
name	varchar(20)	Nombre del grupo
description	varchar(100)	Descripción del grupo

Tabla 4-16: Tabla groups

Fuente: Parte de la Estructura del módulo IonAuth para CodeIgniter

4.2. Diseños del prototipo

4.2.1. Arquitectura del software

Al ser un prototipo desarrollado en el framework CodeIgniter, la arquitectura está basada en el patrón de diseño MVC.

A continuación, se muestra el flujo de una aplicación desarrollada en CodeIgniter:

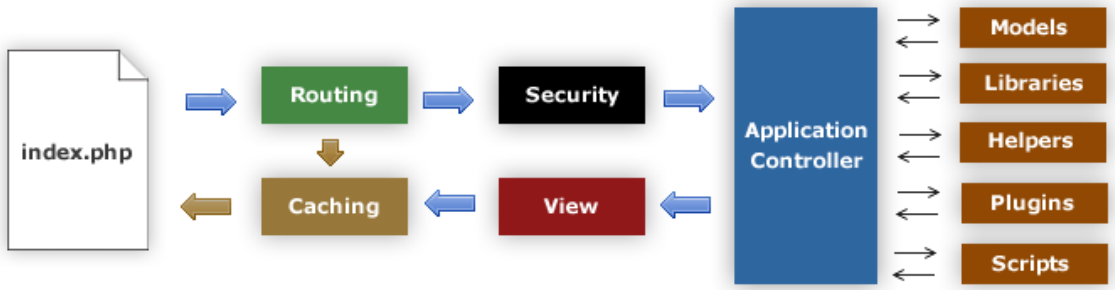


Ilustración 4-3: Flujo de una Aplicación en CodeIgniter

Fuente: (CodeIgniter, 2015) Application FlowChart

4.2.2. Diagramas de secuencia

Se generaron diagramas de secuencia para las acciones de crear, modificar, eliminar de manera genérica, ya que el mismo diagrama aplica para los diferentes requerimientos del prototipo.

4.2.2.1. Crear elementos

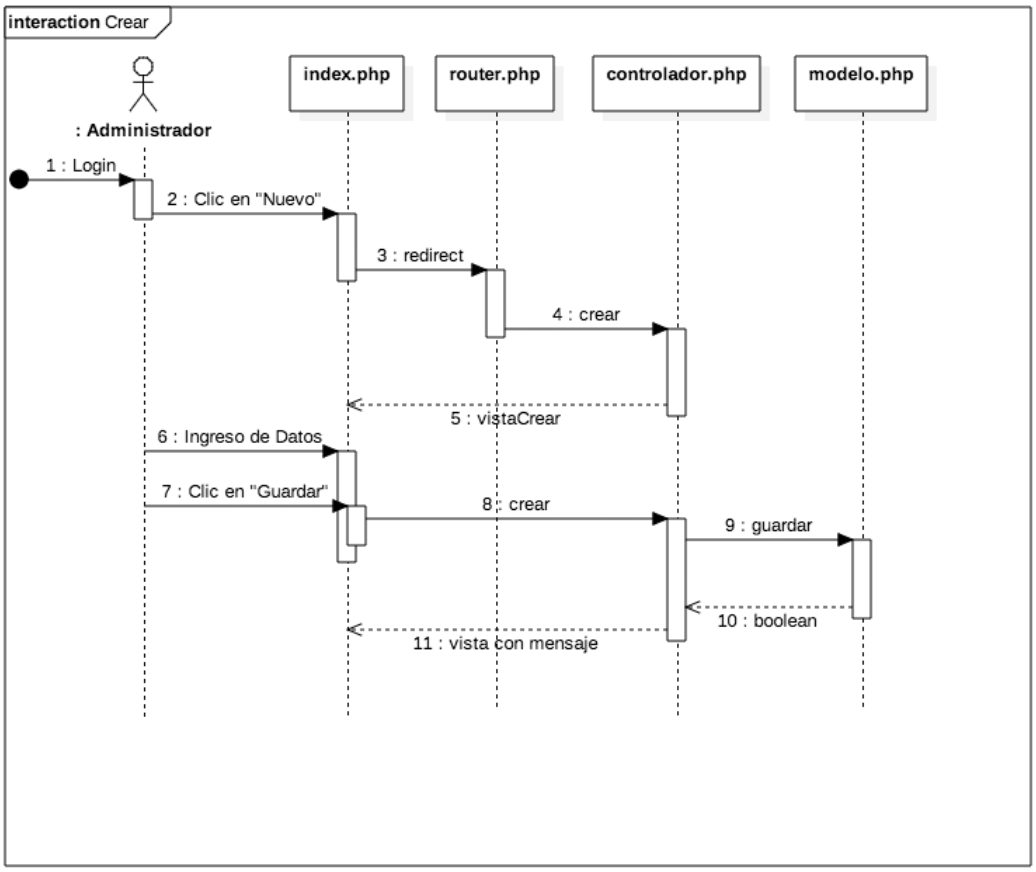


Ilustración 4-4: Diagrama de secuencia para la creación de elementos

Fuente: Autor del proyecto de disertación

4.2.2.2. Modificar elementos

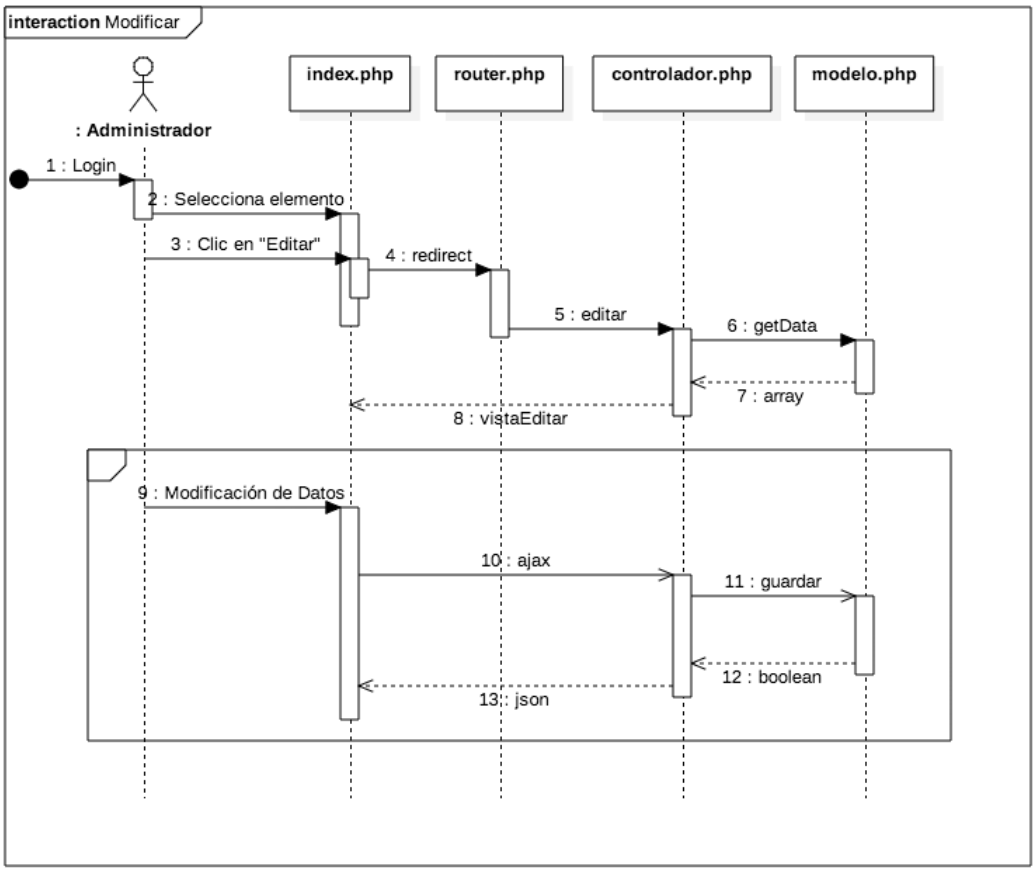


Ilustración 4-5: Diagrama de secuencia para la modificación de elementos

Fuente: Autor del proyecto de disertación

4.2.2.3. Eliminar elementos

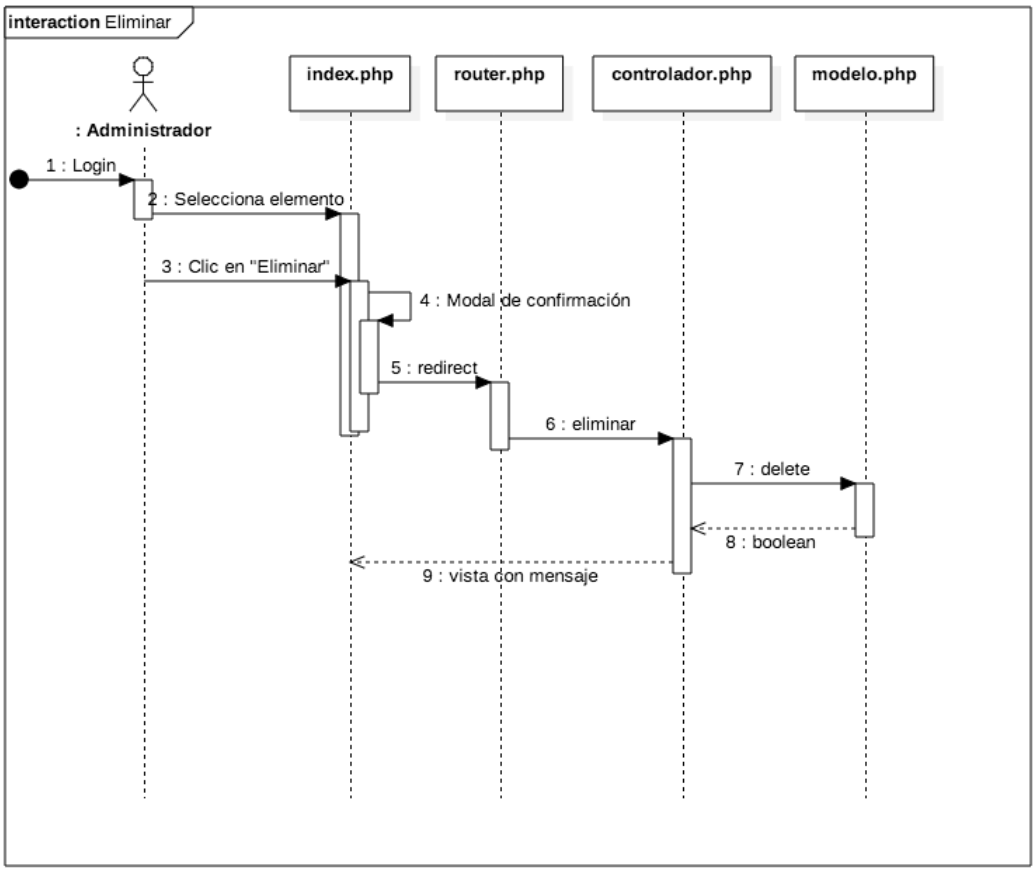


Ilustración 4-6: Diagrama de secuencia para la eliminación de elementos

Fuente: Autor del proyecto de disertación

4.2.2.4. Facturación

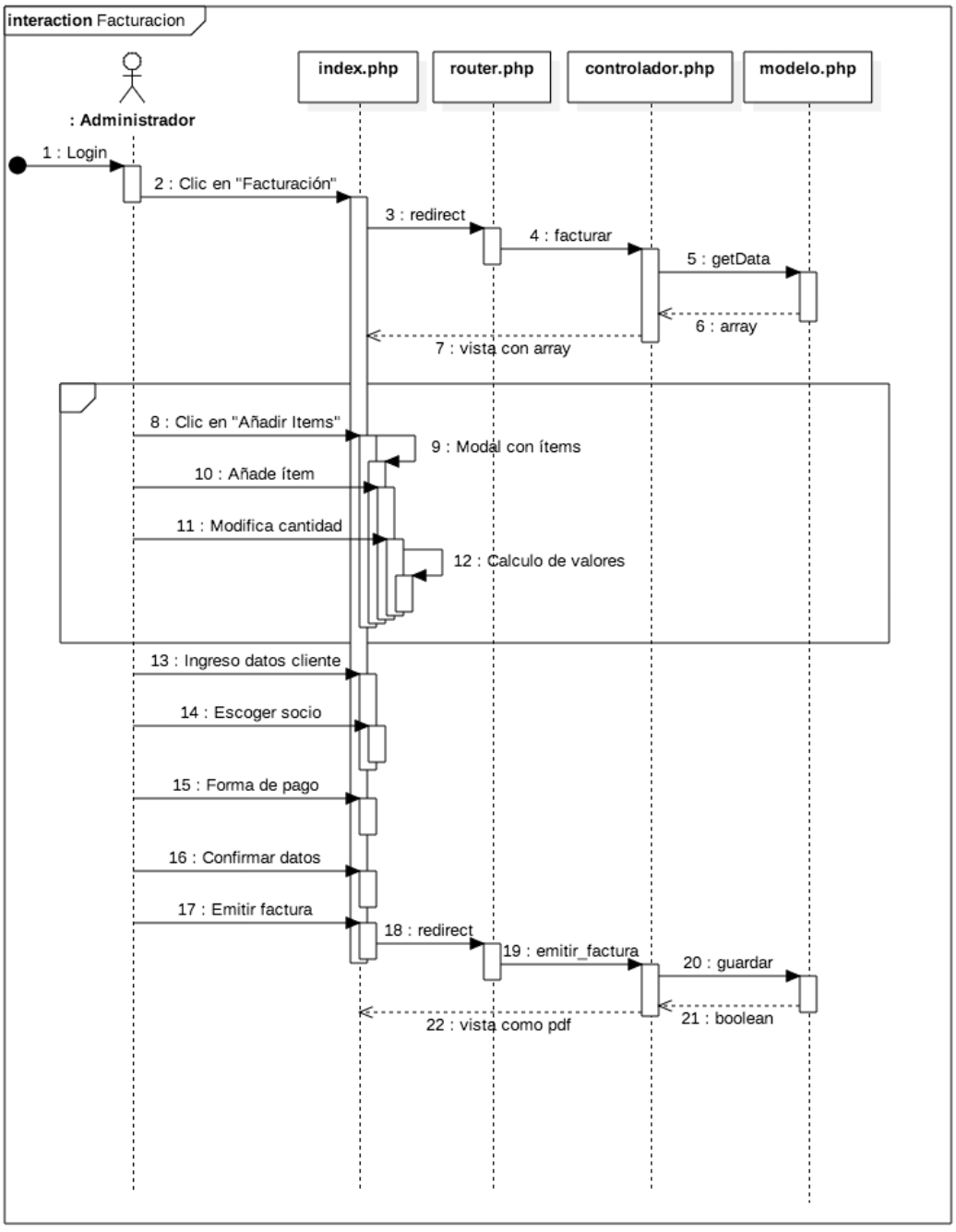


Ilustración 4-7: Diagrama de secuencia para la facturación

Fuente: Autor del proyecto de disertación

4.3. Diseño de interfaces

El prototipo consta de 6 interfaces generales.

A continuación se presentan estas interfaces con su respectiva descripción de funcionalidad:

4.3.1. Login

Formulario para ingresar usuario y contraseña para poder acceder al sistema.



Ilustración 4-8: Interface de Ingreso

Fuente: Autor del proyecto de disertación de grado

4.3.2. Dashboard

Resumen de socios, ventas y comisiones. Acceso solo para usuarios “Gerencia”.



Ilustración 4-9: Interface del Dashboard

Fuente: Autor del proyecto de disertación de grado

4.3.3. Red

Escoge el socio para poder ver su red. Hay un resumen de cada socio de la red. A esta interfase puede acceder el socio y el administrador.

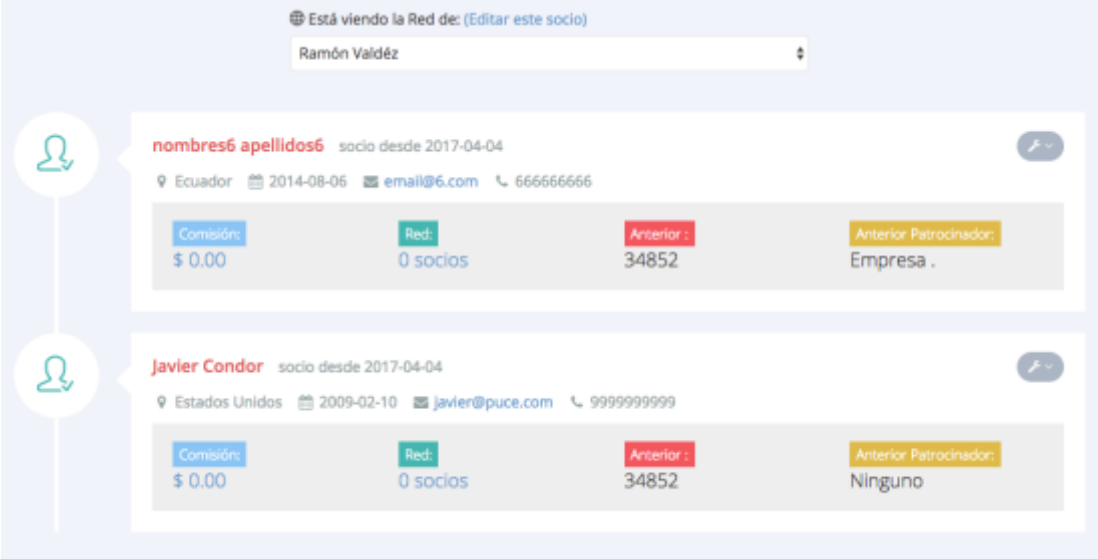


Ilustración 4-10: Interface de la Red de un Socio

Fuente: Autor del proyecto de disertación de grado

4.3.4. Socios

Reporte o listado de los socios de la empresa. Columnas que se muestran: cédula, nombres, apellidos, email, patrocinador (upline), estado, y las acciones de administración (editar, red y comisiones). A esta lista accede el usuario administrador, el mismo que puede editar la información del socio.

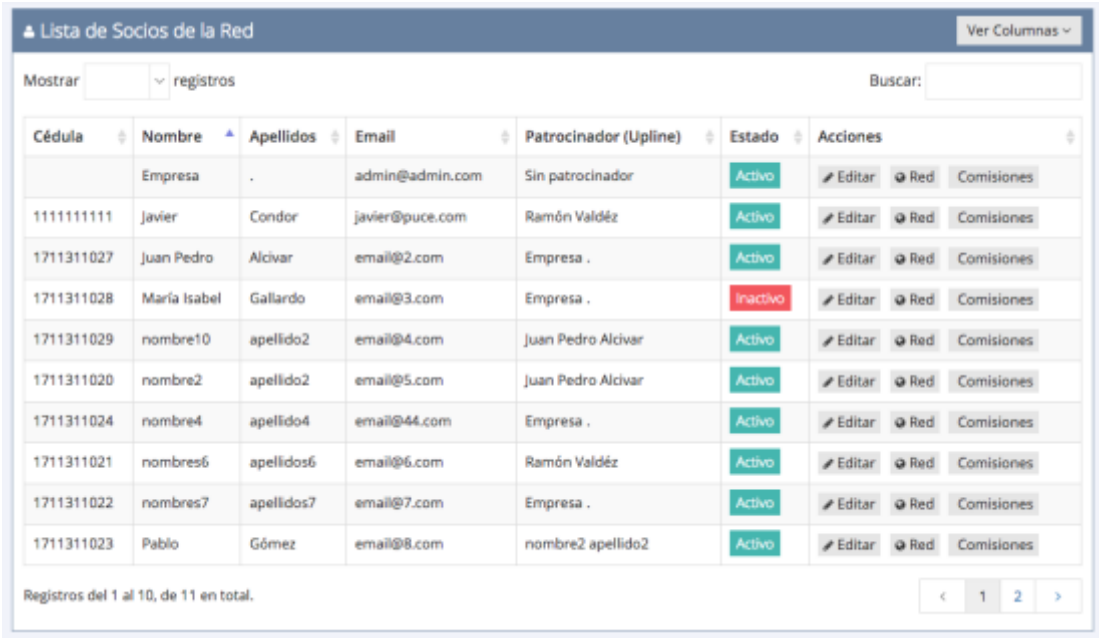


Ilustración 4-11: Interface Socios

Fuente: Autor del proyecto de disertación de grado

4.3.5. Items

Reporte o listado de los items que se pueden facturar. A la intrfase de Items tiene acceso solo el administrador.

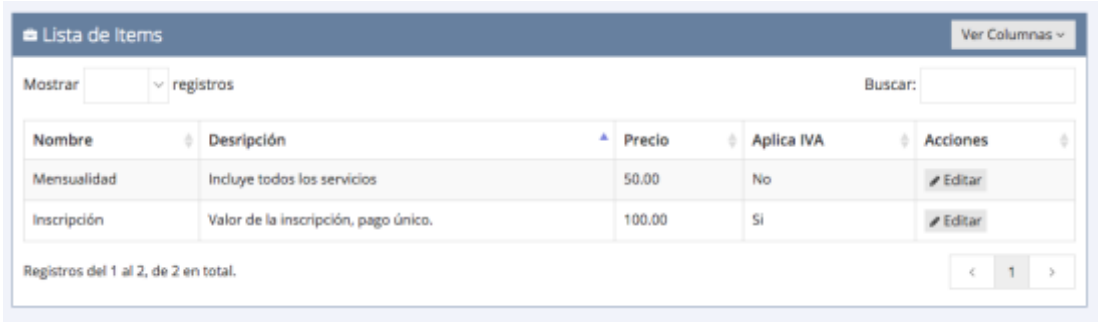


Ilustración 4-12: Interface de Items

Fuente: Autor del proyecto de disertación de grado

4.3.6. Facturación

Esta interfase tiene la modalidad de “asistente”, es decir, va guiando al usuario por 4 pasos para la emisión de la factura:

1. Agregar Items. Agrega los items a facturar
2. Datos de la Factura. Se completan los datos del cliente.
3. Forma de Pago. Se incluye como forma de pago las comisiones acumuladas por el socio.
4. Confirmar. Se verifican los datos ingresados y se procede a la emisión de la factura.

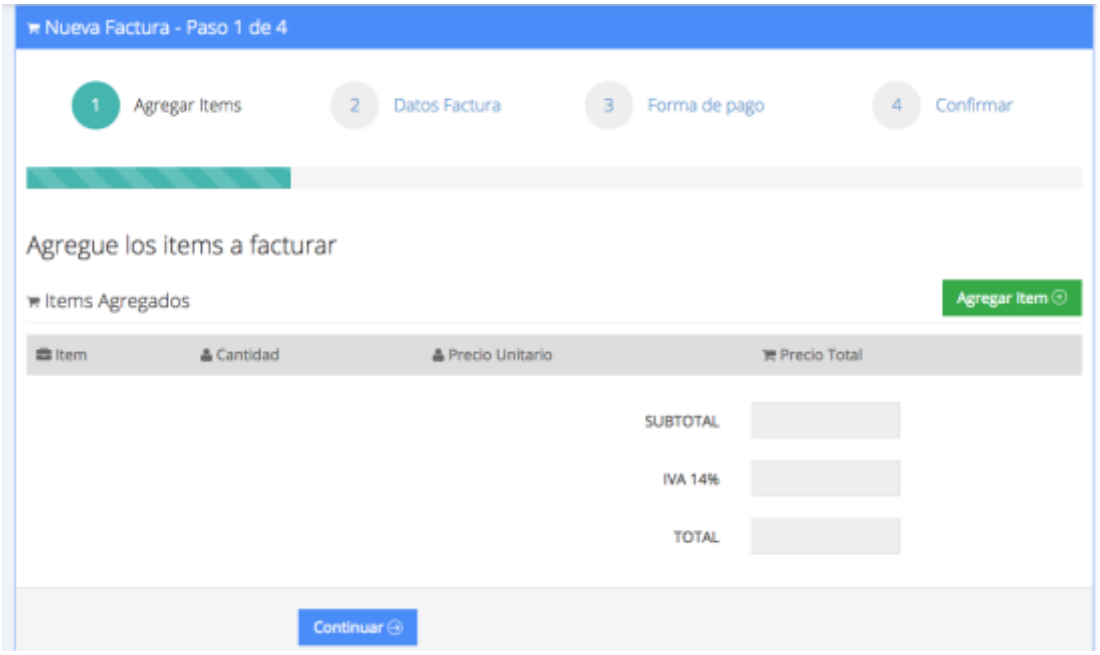


Ilustración 4-13: Interface de Facturación

Fuente: Autor del proyecto de disertación de grado

4.3.7. Parámetros

Reporte o listado de los parámetros que rigen el sistema.



Ilustración 4-14: Interface de Parámetros

Fuente: Autor del proyecto de disertación de grado

CAPÍTULO 5. IMPLEMENTACIÓN Y PRUEBAS

5.1. Plan de iteraciones

En la metodología XP, la planificación de las iteraciones se las plasma en forma de una tabla tomando como entrada las historias de usuario elaboradas previamente.

Iteración	Actividad	Semanas							
		1	2	3	4	5	6	7	8
1	H.1. Gestionar Socios	■							
	H.2. Gestionar Red de Socios		■						
	H.3. Obtener Red			■					
	H.4 Obtener Comisiones				■				
2	H.5. Comprobante de pago					■			
	H.6. Gestionar Items						■		
	H.7. Gestionar Formas de Pago						■		
3	H.8. Gestionar Países							■	
	H.9. Gestionar Configuraciones							■	
4	H.10. Obtener estadísticas								■

Tabla 5-1: Planificación de iteraciones

Fuente: Autor del proyecto de disertación

5.2. Pruebas del prototipo

Según la metodología *Extreme Programming*, las pruebas deben ser unitarias, por cada historia de usuario.

A continuación, se muestran las pruebas realizadas para cada historia de usuario, con los resultados esperados y si el prototipo cumple o no cumple dicho resultado.

5.2.1. Gestionar socios

Historia de Usuario	Actividad	Resultado Esperado	Cumple
H1.1	Crear nuevo socio		
	• Ingreso correcto	Mensaje de éxito	SI
	• Falló el ingreso	Mensaje de error	SI
H1.2	Modificar socio		
	• Modificación correcta	Mensaje de éxito	SI
	• Falló la modificación	Mensaje de error	SI
H1.3	Eliminar socio		
	• Borrado correcto	Mensaje de éxito	SI
	• Falló el borrado	Mensaje de error	SI
H1.4	Listar socios		
	• Listador correcto	Lista de socios	SI
	• Falló el listado	Mensaje de error	SI

Tabla 5-2: Pruebas unitarias para la gestión de socios

Fuente: Autor del proyecto de disertación

5.2.2. Gestionar y obtener red de un socio

Historia de Usuario	Actividad	Resultado Esperado	Cumple
H2	Modificar red de un socios		
	• Modificación correcta	Mensaje de éxito	SI
	• Falló la modificación	Mensaje de error	SI
H3	Obtener red de un socio		
	• Listado correcto	Lista red del socio	SI
	• Falló el listado	Mensaje de error	SI

Tabla 5-3: Pruebas unitarias para la gestión y obtención de la red de un socio

Fuente: Autor del proyecto de disertación

5.2.3. Obtener comisiones de un socio

Historia de Usuario	Actividad	Resultado Esperado	Cumple
H4	Obtener comisiones		
	• Listado correcto	Lista comisiones del socio	SI
	• Falló el listado	Mensaje de error	SI

Tabla 5-4: Pruebas unitarias para la obtención de las comisiones de un socio

Fuente: Autor del proyecto de disertación

5.2.4. Obtener comprobante de pago

Historia de Usuario	Actividad	Resultado Esperado	Cumple
H5.1	Agregar ítems		
	• Agregado correcto	Item agregado	SI
	• Pasar al siguiente paso sin agregar ítems	Mensaje de error	SI
H5.2	Ingreso de datos del cliente		
	• Datos correctos	Pasa al siguiente paso (forma de pago)	SI
	• Datos incorrectos	Mensaje de error	SI
H5.3	Ingreso de la forma de pago		
	• Datos correctos	Pasa al siguiente paso (confirmación)	SI
	• Datos incorrectos	Mensaje de error	SI
H5.4	Emitir comprobante		
	• Datos validados	Comprobante en PDF	SI
	• Datos incorrectos	Mensaje de error	SI

Tabla 5-5: Pruebas unitarias para obtener un comprobante de pago

Fuente: Autor del proyecto de disertación

5.2.5. Gestionar ítems

Historia de Usuario	Actividad	Resultado Esperado	Cumple
H6.1	Crear nuevo ítem		
	• Ingreso correcto	Mensaje de éxito	SI
	• Falló el ingreso	Mensaje de error	SI
H6.2	Modificar ítem		
	• Modificación correcta	Mensaje de éxito	SI
	• Falló la modificación	Mensaje de error	SI
H6.3	Eliminar ítem		
	• Borrado correcto	Mensaje de éxito	SI
	• Falló el borrado	Mensaje de error	SI
H6.4	Listar ítems		
	• Listador correcto	Lista de ítems	SI
	• Falló el listado	Mensaje de error	SI

Tabla 5-6: Pruebas unitarias para la gestión de ítems

Fuente: Autor del proyecto de disertación

5.2.6. Gestionar formas de pago

Historia de Usuario	Actividad	Resultado Esperado	Cumple
H7.1	Crear nueva forma de pago		
	• Ingreso correcto	Mensaje de éxito	SI
	• Falló el ingreso	Mensaje de error	SI
H7.2	Modificar forma de pago		
	• Modificación correcta	Mensaje de éxito	SI
	• Falló la modificación	Mensaje de error	SI
H7.3	Eliminar forma de pago		
	• Borrado correcto	Mensaje de éxito	SI
	• Falló el borrado	Mensaje de error	SI
H7.4	Listar forma de pago		
	• Listador correcto	Lista de formas de pago	SI
	• Falló el listado	Mensaje de error	SI

Tabla 5-7: Pruebas unitarias para la gestión de formas de pago

Fuente: Autor del proyecto de disertación

5.2.7. Gestionar países

Historia de Usuario	Actividad	Resultado Esperado	Cumple
H8.1	Crear nuevo país		
	• Ingreso correcto	Mensaje de éxito	SI
	• Falló el ingreso	Mensaje de error	SI
H8.2	Modificar país		
	• Modificación correcta	Mensaje de éxito	SI
	• Falló la modificación	Mensaje de error	SI
H8.3	Eliminar país		
	• Borrado correcto	Mensaje de éxito	SI
	• Falló el borrado	Mensaje de error	SI
H8.4	Listar países		
	• Listador correcto	Lista de países	SI
	• Falló el listado	Mensaje de error	SI

Tabla 5-8: Pruebas unitarias para la gestión de países

Fuente: Autor del proyecto de disertación

5.2.8. Gestionar configuraciones del sistema

Historia de Usuario	Actividad	Resultado Esperado	Cumple
H9.1	Crear nueva variable		
	• Ingreso correcto	Mensaje de éxito	SI
	• Falló el ingreso	Mensaje de error	SI
H9.2	Modificar variable		
	• Modificación correcta	Mensaje de éxito	SI
	• Falló la modificación	Mensaje de error	SI
H9.3	Eliminar variable		
	• Borrado correcto	Mensaje de éxito	SI
	• Falló el borrado	Mensaje de error	SI
H9.4	Listar variables		
	• Listador correcto	Lista de variables	SI
	• Falló el listado	Mensaje de error	SI

Tabla 5-9: Pruebas unitarias para la gestión de configuraciones del sistema

Fuente: Autor del proyecto de disertación

5.2.10. Obtener estadísticas

Historia de Usuario	Actividad	Resultado Esperado	Cumple
H10.1	Obtener estadísticas		
	• Consultas correctas	Estadísticas desplegadas	SI
	• Falló las consultas	Mensaje de error	SI

Tabla 5-10: Pruebas unitarias para la obtención de estadísticas

Fuente: Autor del proyecto de disertación

CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES

El objetivo general de este proyecto es desarrollar un software prototipo que administre una red de mercadeo con un sistema de compensación uninivel simple y compactación de socios, en plataforma web, utilizando software libre, el mismo que ha concluido con éxito. A continuación, se presentan las conclusiones y recomendaciones:

6.1. Conclusiones

- Es vital para el proceso de levantamiento de requerimientos la selección de una adecuada técnica para tal acción, tomando en cuenta las ventajas y desventajas de cada una de estas técnicas o metodologías. El uso de la entrevista como herramienta permitió una correcta comprensión de los requerimientos en cuanto al esquema uni nivel simple con compactación.

- La selección de la metodología *Extreme Programming* para el desarrollo de este proyecto contribuyó para que la construcción del prototipo sea ágil, sin estancarse en la burocracia, a veces innecesaria, generada por otras metodologías.

- Hoy en día, el uso de la plataforma web para la construcción de sistemas de información es imprescindible, ya que brinda al usuario información precisa en tiempo real. Para sistemas de mercadeo en red, el tener una plataforma que esté en la web, es casi una obligación, ya que todos sus socios necesitan información de su red de manera rápida y precisa.

- El uso de frameworks y librerías de terceros hacen que el proyecto sea más ágil en la etapa de desarrollo, obteniendo reducción en los tiempos establecidos y una garantía de funcionamiento implícita.

- A además del uso del mencionado framework y librerías para el desarrollo de sistemas basados en el patrón de diseño MVC bajo PHP, es necesario tener bases sólidas de desarrollo con programación orientada a objetos para solventar cualquier cambio que dichas librerías requieran, o para personalizar las funcionalidades según los requerimientos levantados.
- El desarrollo modular y genérico del sistema permite una futura adaptación a cualquier negocio que desee implementar el sistema de compensación mencionado en esta disertación.
- La usabilidad de un sistema es parte fundamental del desarrollo de un sistema web, y mucho más relevante si usuarios no técnicos van a hacer uso periódico del mismo.

6.2. Recomendaciones

- Al usar librerías externas, verificar no solo la funcionalidad sino también si existe documentación detallada, si se encuentra en constante evolución (nuevas versiones periódicas), y si son de una fuente confiable. Así, el sistema tendrá cimientos sólidos y garantizados.
- Desarrollar nuevos módulos que complementen el prototipo desarrollado, como por ejemplo facturación electrónica, módulos contables, etc. para que la versión del software pase de ser un prototipo a una versión final de un producto que pueda tener uso práctico.
- Modificar el prototipo para que no solo maneje un sistema de

compensación (uninivel simple con compactación) sino que acepte varios para diversificar su uso.

- En el caso de que se desee continuar con el desarrollo, es altamente recomendable que se actualicen los componentes usados en el presente prototipo, haciendo pruebas visuales y de funcionalidad para que el sistema conserve su objetivo principal.

GLOSARIO DE TÉRMINOS

A

AEVD

Asociación Ecuatoriana de Venta Directa ..... VIII, 17, 21, 22, 23, 24, 49

AJAX

Asinchronous Javascript and XML. Técnica de desarrollo para ejecutar comunicaciones entre el cliente y el servidor en segundo plano. ..... V, 10, 38, 39

C

C

Lenguaje de programación..... 35, 37

CERN

Organización Europea para la Investigación Nuclear 32

CSS

Cascade Style Sheets. Hojas de estilos en cascada. 10, 39

D

DOM

Document Object Model..... 38, 39

F

Facebook

Red social fundada por Mark Zuckerberg y Eduardo Saverin en el 2004. Web www.facebook.com..... 39

H

HTML

HyperText Markup Language. Lenguaje de Marcado de HyperTexto. .. V, VII, VIII, 10, 31, 32, 33, 35, 36, 39, 94, 95, 96, 97, 98, 99

I

IETF
Grupo de Trabajo de Ingeniería de Internet 32

J

Java
Lenguaje de programación 35

L

lenguaje de scripting
lenguaje de programación interpretado 38, 39

M

MLM
Multi Level Marketing. Marketing Multinivel. 21
Multilevel Marketing. Marketing Multinivel ..... IV, 49
motor Zend
es un motor de procesamiento para la intepretación y cifrado del código php, desde la versión 4.
Desarrollado por Zend Technologies para brindar un equipo de soporte y acelerar la carga de aplicaciones realizadas con php 37
MVC
Modelo-Vista-Controlador. Es un patron de diseño de software. 10
Patrón de diseño Modelo - Vista - Controlador 40, 43, 84
MySQL
Motor de Base de Datos 37

P

Perl
Lenguaje de programación 35, 36
PHP
Acrónimo de "PHP
Hypertext Preprocessor", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser incrustado en páginas HTML 31, 91

PHP, acrónimo de "PHP
Hypertext Preprocessor", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser incrustado en páginas HTML. ..... V, 10, 35, 36, 37, 84

S

sistema de compensación uninivel con compactación de socios
Uninivel híbrido o uninivel con compresión dinámica, o uninivel con compactación. 45
SOAP
protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. 37

W

W3C
World Wide Web Consortium 91
World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web ..... VIII, 32

X

XHTML
Combina las características de HTML y XML y es considerado como un puente entre estas dos. 39

REFERENCIAS BIBLIOGRÁFICAS

Comisión Federal de Comercio de USA. (26 de Octubre de 2015). *Información para Consumidores*. Obtenido de Planes de comercialización multiniveles: http://www.consumidor.ftc.gov/articulos/s0065-planes-de-comercializacion-multiniveles

Asociación Ecuatoriana de Venta Directa, AEVD. (26 de Octubre de 2015). *Ley de Defensa del Consumidor*. Obtenido de DERECHOS Y OBLIGACIONES DEL CONSUMIDOR: http://www.aevd.ec/index.php?option=com_content&view=article&id=75&Itemid=83

Asociación Ecuatoriana de Venta Directa, AEVD. (26 de Octubre de 2015). *Nosotros*. Obtenido de HISTORIA DE LA VENTA DIRECTA: http://www.aevd.ec/index.php?option=com_content&view=article&id=91&Itemid=99

Diaz M., J. A. (09 de Enero de 2010). *Multinivel*. Obtenido de Breve Historia del Marketing Multinivel: http://www.articulo.org/articulo/11058/breve_historia_del_marketing_multinivel.html

Samartano, G. (26 de Octubre de 2015). *La Revolucion Económica se está gestando*. Obtenido de Historia del Marketing Multinivel: http://gonzalo-samartano.blogspot.com/p/historia-del-marketing-multinivel.html

NUTRILITE. (26 de Octubre de 2015). *Why Nutrilite*. Obtenido de History: http://www.nutrilite.com/en-us/nature/whynutrilite/our-history.aspx

Direct Selling News. (04 de Junio de 2015). *DSN Global 100*. Obtenido de 2015 Profiles: http://directsellingnews.com/index.php/view/2015_profiles#.Vi3htrcvfIU

Wikipedia. (26 de Octubre de 2015). *Marketing Multinivel*. Obtenido de Planes de compensación: https://es.wikipedia.org/wiki/Marketing_multinivel

Wikipedia. (26 de 10 de 2015). *HTML*. Obtenido de HyperText Markup Language: Fuente: http://es.wikipedia.org/wiki/HTML

Franganillo, J. (06 de Septiembre de 2010). *Anuario ThinkEPI.* Obtenido de Html5: el nuevo estándar básico de la Web: http://franganillo.es/html5.pdf

Wikipedia. (26 de Octubre de 2015). *HTML5*. Obtenido de HyperText Markup Language, versión 5: Fuente: http://es.wikipedia.org/wiki/HTML5

W3Schools. (26 de Octubre de 2015). *HTML Element Reference*. Obtenido de HTML Tags Ordered by Category: http://www.w3schools.com/tags/ref_byfunc.asp

PHP. (26 de Octubre de 2015). *Manual de PHP*. Obtenido de Apéndices - Historia de PHP y Proyectos Relacionados: http://php.net/manual/es/history.php.php

Crockford, D. (26 de Octubre de 2015). *JavaScript: The World's Most Misunderstood Programming Language*. Obtenido de http://www.crockford.com/javascript/javascript.html

Wikipedia. (26 de Octubre de 2015). *Javascript*. Obtenido de Uso en páginas web: https://es.wikipedia.org/wiki/JavaScript

EllisLab. (26 de Octubre de 2015). *About - CodeIgniter*. Obtenido de A Brief History of CodeIgniter: https://ellislab.com/codeigniter

CodeIgniter. (26 de Octubre de 2015). *Documentation - Server Requirements*. Obtenido de Server Requirements: http://www.codeigniter.com/user_guide/general/requirements.html

CodeIgniter. (26 de Octubre de 2015). *Documentation - Application Flow Chart*. Obtenido de Application Flow Chart: http://www.codeigniter.com/user_guide/overview/appflow.html

Edmunds, B. (26 de Octubre de 2015). *Ion Auth 2 - Usage*. Obtenido de Usage: https://github.com/benedmunds/CodeIgniter-Ion-Auth

Martínez Guerrero , J. M., & Silva Delgado, C. A. (25 de 01 de 2011). Guía Metodológica. *Levantamiento y análisis de Requerimientos*. Bogotá D.C., Cundinamarca, Colómbia: PONTIFICIA UNIVERSIDAD JAVERIANA, FACULTAD DE INGENIERIA.

Aristizábal Mejía, N., & Torres Moreno, M. E. (2009). Técnicas de Levantamiento de Requerimientos con Innovación. *Cuarto Congreso Colombiano de Computación 4CCC Sociedad Colombiana de Computación S(Co)2* (pág. 2). Bucaramanga: Universidad Autónoma de Bucaramanga UNAB.

Herrera J., L. J. (26 de Octubre de 2015). *Monografias.com*. Obtenido de Ingeniería De Requerimientos Ingeniería De Software: http://www.monografias.com/trabajos6/resof/resof2.shtml

Ponguillo Rodríguez, E. (2015). *Investigación sobre estándares de red.* Guayaquil: Escuela Superior Poitécnica del Litoral, ESPOL. MSIG.

Wikipedia. (26 de 10 de 2015). *Motor Zend*. Obtenido de Motor Zend: https://es.wikipedia.org/wiki/Motor_Zend

Wikipedia. (26 de Octubre de 2016). *SGML*. Obtenido de SGML: https://es.wikipedia.org/wiki/SGML

Kiyosaki, R. T. (2004). El Cuadrante de Flujo de Dinero. En R. Kiyosaki, *El Cuadrante de Flujo de Dinero* (pág. 112). Mexico, Mexico: Santillana Ediciones Generales S.A. de C.V.

Beck, K., & Andres, C. (2012). *Extreme Programming Explained: Embrace Change.* Obtenido de http://ptgmedia.pearsoncmg.com/images/9780321278654/samplepages/9780321278654.pdf

Joskowicz, J. (10 de Febrero de 2008). *Reglas y Prácticas en eXtreme Programming.* Obtenido de Reglas y Prácticas en eXtreme Programming: https://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf

GNU Operating System. (21 de Abril de 2017). *What is free software?* Obtenido de GNU Operating System: http://www.gnu.org/philosophy/free-sw.html

Wikipedia. (2017). *XAMPP*. Obtenido de Wikipedia: https://es.wikipedia.org/wiki/XAMPP

Apache Friends. (2017). *Acerca de*. Obtenido de Apache Friends: https://www.apachefriends.org/es/about.html

World Wide Web Consortium. (2017). *W3C HTML*. Obtenido de W3C: https://www.w3.org/html/

Luján Mora, S. (8 de Octubre de 2001). *Programación en Internet: Clientes Web.* Obtenido de Programación en Internet: Clientes Web: https://rua.ua.es/dspace/bitstream/10045/16994/1/sergio_lujan-programacion_en_internet_clientes_web.pdf

World Wide Web Consortium. (26 de Octubre de 2015). *HTML/Training/What is HTML*. Obtenido de History: https://www.w3.org/community/webed/wiki/HTML/Training/What_is_HTML%3F

Mozilla Developer Network. (2017). *HTML5 - HTML*. Obtenido de MDN Mozilla Developer Network: https://developer.mozilla.org/es/docs/HTML/HTML5

PHP. (26 de Octubre de 2015). *Prefacio*. Obtenido de Manual de PHP: http://www.php.net/manual/es/preface.php

StatCounter. (Marzo de 2017). *Browser Market Share Worldwide*. Obtenido de StatCounter - Global Stats: http://gs.statcounter.com/

CodeIgniter. (26 de Octubre de 2015). *Model - View - Controller*. Obtenido de CodeIgniter Documentation - User Guide: https://codeigniter.com/user_guide/

Oracle Corporation. (2017). *What is MySQL*. Obtenido de MySQL Reference Manual: https://dev.mysql.com/doc/refman/5.5/en/what-is-mysql.html

Oracle Corporation. (2017). *General Information*. Obtenido de MySQL Reference Manual: https://dev.mysql.com/doc/refman/5.5/en/introduction.html

World Wide Web Consortium - Oficina española. (2017). *Guía breve de CSS*. Obtenido de Sitio web de la oficina española de W3C: http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo

Hanan Tek. (25 de Junio de 2010). *Documentancion Generada en Extreme Programming*. Obtenido de Hanan Tek: http://www.hanantek.com/documentacion-programacion-extrema

ANEXOS

A continuación, se detallan los anexos incluidos en el CD:

ANEXO 1 – Diagramas de secuencias

ANEXO 2 – Esquemas conceptual y lógico

ANEXO 3 – Código fuente y script SQL

ANEXO 4 – Herramientas y software utilizado

ANEXO 5 – Manuales y tablas de referencia