



ESPE

**UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA**

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA CARRERA DE INGENIERÍA EN SOFTWARE

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SOFTWARE**

TEMA:

**“DESARROLLO DE UN SITIO WEB BAJO EL PATRÓN DE
DISEÑO MODELO, VISTA, CONTROLADOR, QUE PERMITA
LA GESTIÓN DEL REGISTRO DE VOLUNTARIOS Y
DONACIONES PARA DAR SOPORTE E INFORMACIÓN A LA
FUNDACIÓN CONSTRUYAMOS UN SUEÑO, UBICADA EN LA
CIUDAD DE LATACUNGA EN EL AÑO 2015”**

AUTOR:

DANIEL ALEJANDRO OBANDO MOSQUERA

DIRECTORA:

ING. IVONE ARIAS

LATACUNGA

2015



**DEPARTAMENTO ELÉCTRICA Y ELECTRÓNICA
CARRERA INGENIERÍA EN SOFTWARE**

CERTIFICACIÓN

Certifico que el trabajo de titulación “**DESARROLLO DE UN SITIO WEB BAJO EL PATRÓN DE DISEÑO MODELO, VISTA, CONTROLADOR, QUE PERMITA LA GESTIÓN DEL REGISTRO DE VOLUNTARIOS Y DONACIONES PARA DAR SOPORTE E INFORMACIÓN A LA FUNDACIÓN CONSTRUYAMOS UN SUEÑO, UBICADA EN LA CIUDAD DE LATACUNGA EN EL AÑO 2015**” realizado por el señor **OBANDO MOSQUERA DANIEL ALEJANDRO**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor **DANIEL ALEJANDRO OBANDO MOSQUERA** para que lo sustente públicamente.

Latacunga, 04 de Agosto del 2016

Ing. Ivone Arias Almeida.

DIRECTOR



**DEPARTAMENTO ELÉCTRICA Y ELECTRÓNICA
CARRERA INGENIERÍA EN SOFTWARE**

AUTORÍA DE RESPONSABILIDAD

AUTOR: DANIEL ALEJANDRO OBANDO MOSQUERA

Declaro, que el trabajo denominado: **“DESARROLLO DE UN SITIO WEB BAJO EL PATRÓN DE DISEÑO MODELO, VISTA, CONTROLADOR, QUE PERMITA LA GESTIÓN DEL REGISTRO DE VOLUNTARIOS Y DONACIONES PARA DAR SOPORTE E INFORMACIÓN A LA FUNDACIÓN CONSTRUYAMOS UN SUEÑO, UBICADA EN LA CIUDAD DE LATACUNGA EN EL AÑO 2015”**, se desarrolló en base a investigación, respetando derechos intelectuales y realizando las citas correspondientes incorporadas en la bibliografía.

Consecuentemente este trabajo es de mi autoría y me responsabilizo del contenido y veracidad del proyecto en mención.

Latacunga, 24 de Febrero del 2016

DANIEL ALEJANDRO OBANDO MOSQUERA

C.I.: 0401299524



**DEPARTAMENTO ELÉCTRICA Y ELECTRÓNICA
CARRERA INGENIERÍA EN SOFTWARE**

AUTORIZACIÓN

Yo, Daniel Alejandro Obando Mosquera Autorizo a la Universidad de las Fuerzas Armadas ESPE Latacunga la publicación, en la biblioteca virtual de la Institución el trabajo **“DESARROLLO DE UN SITIO WEB BAJO EL PATRÓN DE DISEÑO MODELO, VISTA, CONTROLADOR, QUE PERMITA LA GESTIÓN DEL REGISTRO DE VOLUNTARIOS Y DONACIONES PARA DAR SOPORTE E INFORMACIÓN A LA FUNDACIÓN CONSTRUYAMOS UN SUEÑO, UBICADA EN LA CIUDAD DE LATACUNGA EN EL AÑO 2015”**, cuyo contenido, ideas y criterios son de exclusiva responsabilidad y autoría.

Latacunga, 24 de Febrero del 2016

DANIEL ALEJANDRO OBANDO MOSQUERA

C.I.: 0401299524

DEDICATORIA

Dedico este trabajo a mis padres Silvana, Daniel, a mi hijo Mateo el motor de mi inspiración, también a una persona que es muy importante en esta etapa de mi vida, Gissela por creer en mí por haberme apoyado durante mi carrera universitaria a Camilo un gran amigo por haber compartido conmigo largas horas de estudio y amistad.

Alejandro Obando

AGRADECIMIENTO

Mi agradecimiento va dirigido a quien con su esfuerzo, amor y dedicación supieron alentarme a seguir adelante y no decaer mis padres, también quiero agradecer a la Universidad de las Fuerzas Armadas - ESPE quien me acogió y permitió engrandecer mi espíritu por la investigación, mediante el conocimiento y valores que sus profesores impartieron en mí, gracias a ellos pude realizar la culminación de esta pequeña etapa de mi vida, quiero agradecer también infinitamente a la Ingeniera Ivone Arias por su colaboración y constante paciencia en la guía de este proyecto, muchas gracias de corazón a mis queridos profesores, que en el camino como estudiante supieron transmitir no solamente conocimiento, sino también muchos valores éticos y morales para aplicarlos en la sociedad, a cada uno de ellos les tengo un gran respeto y siempre los recordaré en mi carrera como profesional, también quiero agradecer a varios amigos Cristhian, Juanca, David por darme sus manos sin recibir nada a cambio y por ultimo mi familia mi tía Soraya, mi tío Pablo y mi abuelita que dejaron una huella en mi dándome su apoyo siempre.

Alejandro Obando.

ÍNDICE DE CONTENIDO

| | |
|---|-------------|
| PORTADA | i |
| CERTIFICACIÓN | ii |
| AUTORÍA DE RESPONSABILIDAD | iii |
| AUTORIZACIÓN | iv |
| DEDICATORIA | v |
| AGRADECIMIENTO | vi |
| ÍNDICE DE CONTENIDO | vii |
| ÍNDICE DE FIGURAS | xii |
| ÍNDICE DE TABLAS | xiii |
| RESUMEN | xv |
| ABSTRACT | xvi |

CAPÍTULO I

| | |
|---|----------|
| 1. INTRODUCCIÓN AL PROYECTO | 1 |
| 1.1. Antecedentes | 1 |
| 1.2. Planteamiento y formulación del problema | 3 |
| 1.3. Justificación e importancia | 3 |
| 1.4. Objetivos generales y específicos..... | 4 |
| 1.4.1. Objetivo General | 4 |
| 1.4.2. Objetivos Específicos..... | 4 |
| 1.5. Metas | 5 |

CAPÍTULO II

| | |
|---|----------|
| 2. MARCO TEÓRICO | 6 |
| 2.1. Introducción del capítulo | 6 |
| 2.2. Sistemas basados en técnicas WEB. | 6 |
| 2.2.1. Diferencia entre un sitio y una página web. | 7 |
| 2.3. Definición de ingeniería web | 8 |

| | |
|--|----|
| 2.3.1. Proceso de la ingeniería web..... | 9 |
| 2.3.2. Formulación | 9 |
| 2.3.3. Planificación..... | 10 |
| 2.3.4. Modelado | 11 |
| 2.3.5. Generación de páginas | 11 |
| a. Diseño de la interfaz..... | 11 |
| b. Diseño de las páginas..... | 12 |
| 2.3.6. Test y evaluación | 12 |
| 2.4. Lenguajes de programación web..... | 13 |
| 2.4.1. Lenguaje ASP o ASP.net..... | 14 |
| 2.4.2. HTML | 15 |
| a. Ventajas..... | 15 |
| b. Desventajas | 16 |
| 2.4.3. PHP | 16 |
| a. Ventajas..... | 16 |
| b. Desventajas..... | 17 |
| 2.4.4. Lenguaje Javascript | 18 |
| a. Ventajas..... | 18 |
| b. Desventajas..... | 18 |
| 2.4.5. Lenguaje JSP..... | 19 |
| 2.5. Patrón de diseño Modelo, vista, controlador (MVC)..... | 20 |
| 2.5.1. Modelo | 21 |
| 2.5.2. Vista | 21 |
| 2.5.3. Controlador | 22 |
| 2.5.4. Ventajas y desventajas del Patrón Modelo Vista – Controlador..... | 22 |
| a. Ventajas..... | 23 |
| b. Desventajas | 23 |
| 2.6. Framework..... | 23 |
| 2.6.1. Características principales..... | 24 |
| 2.6.2. Symfony | 24 |
| 2.7. Tipos de Framework WEB | 25 |

| | |
|--|----|
| 2.7.1. Codeigniter..... | 26 |
| 2.7.2. Yii Framework | 27 |
| 2.8. Bases de datos | 27 |
| 2.8.1. Tipos de bases de Datos | 28 |
| a. MySQL | 28 |
| b. PostgreSQL..... | 29 |
| 2.9. Metodologías ágiles y tradicionales de desarrollo web. | 30 |
| 2.9.1. Metodologías tradicionales | 31 |
| 2.9.2. Metodologías ágiles | 32 |
| a. Metodología XP | 32 |
| b. Metodología Ágil Scrum..... | 39 |

CAPÍTULO III

| | |
|---|-----------|
| 3. SELECCIÓN DE LA TECNOLOGÍA | 50 |
| 3.1. Selección del lenguaje de Programación. | 50 |
| 3.2. Selección Base de Datos | 51 |
| 3.2.1. MySQL | 51 |
| 3.3. Frameworks | 52 |
| 3.3.1. Comparativa de frameworks. | 52 |
| 3.4. Metodologías de desarrollo ágil. | 54 |
| 3.4.1 Comparativa de las metodologías..... | 55 |

CAPÍTULO IV

| | |
|--|-----------|
| 4. DESARROLLO DEL SITIO WEB | 57 |
| 4.1. Utilización de la metodología ágil Scrum en el Proyecto. | 58 |
| 4.1.1. Pila del producto. | 58 |
| 4.1.2. Historias de usuario | 58 |
| 4.1.3. Pila del producto | 65 |
| 4.1.4. Pila del sprint | 66 |
| 4.2. Diseño de los formularios..... | 67 |

| | |
|--|----|
| 4.2.1. Formulario registro de voluntario..... | 67 |
| 4.2.2. Formulario registro de Donaciones..... | 68 |
| 4.2.3. Formulario registro empleados..... | 68 |
| 4.3. Diseño de la base de datos..... | 69 |
| 4.4. Arquitectura del sistema..... | 70 |
| 4.4.1. Modelo (Model)..... | 71 |
| 4.4.2. Vista (View)..... | 71 |
| 4.4.3. Controlador (controller)..... | 72 |
| 4.5. Diseño de las interfaces..... | 75 |
| 4.5.1. Diseño del sitio web..... | 75 |
| 4.6. Implementación..... | 80 |
| 4.7. Codificación Framework Codeigniter (MVC)..... | 82 |
| 4.8. Pruebas del sistema..... | 83 |
| 4.8.1. Pruebas de interfaz y contenido..... | 84 |
| 4.8.2. Pruebas funcionales y operacionales..... | 85 |

CAPÍTULO V

| | |
|---|-----------|
| 5. VALIDACIÓN DE LA PROPUESTA..... | 87 |
| 5.1. Entrega del sistema..... | 87 |
| 5.1.1. Descripción del proyecto..... | 87 |
| 5.1.2. Alcance del proyecto..... | 87 |
| 5.1.3. Supuestos..... | 87 |
| 5.1.4. Restricciones..... | 88 |
| 5.1.5. Requerimientos del proyecto..... | 88 |
| 5.1.6. Requerimientos Generales..... | 88 |
| 5.1.7. Requerimientos Específicos..... | 88 |
| 5.1.8. Requerimientos no funcionales..... | 89 |
| 5.2. Aceptación del sistema..... | 89 |
| 5.3. Manual de usuario..... | 89 |
| 5.4. Manual de configuración del sistema..... | 90 |

| | |
|------------------------------------|----|
| 5.5. Conclusión del capítulo | 90 |
|------------------------------------|----|

CAPÍTULO VI

6. CONCLUSIONES Y RECOMENDACIONES

| | |
|-------------------------|----|
| 6.1. Conclusiones | 91 |
|-------------------------|----|

| | |
|----------------------------|----|
| 6.2. Recomendaciones | 92 |
|----------------------------|----|

| | |
|---|-----------|
| REFERENCIAS BIBLIOGRÁFICAS | 93 |
|---|-----------|

| | |
|--------------------|-----------|
| ANEXOS..... | 99 |
|--------------------|-----------|

Anexo A. Aceptación del sistema

Anexo B. Manual de usuario y configuración

Anexo C. Ficha de voluntario

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura No. 1. Arquitectura de conexión e un sitio web..... | 7 |
| Figura No. 2. Proceso de Ingeniería web..... | 13 |
| Figura No. 3. Arquitectura del Modelo, Vista, Controlador..... | 20 |
| Figura No. 4. Fases XP..... | 35 |
| Figura No. 5. Desarrollo Scrum..... | 40 |
| Figura No. 6. Pila del producto y pila del sprint..... | 42 |
| Figura No. 7. Historia de usuario Scrum..... | 44 |
| Figura No. 8 Pila de producto | 44 |
| Figura No. 9. Ejemplo pila sprint..... | 46 |
| Figura No. 10. Modelo evolutivo iterativo | 47 |
| Figura No. 11. Bases de datos locales | 51 |
| Figura No. 12. Características de Framework MVC..... | 53 |
| Figura No. 13. Diseño de la base de datos..... | 69 |
| Figura No. 14. Arquitectura del sistema..... | 71 |
| Figura No. 15. Diagrama de interfaz común | 75 |
| Figura No. 16. Interfaz de usuario inicial..... | 76 |
| Figura No. 17. Interfaz historia de la fundación | 77 |
| Figura No. 18. Interfaz proyectos sociales..... | 77 |
| Figura No. 19. Interfaz proyectos culminados..... | 78 |
| Figura No. 20. Interfaz registro voluntario..... | 78 |
| Figura No. 21. Interfaz registro donación..... | 79 |
| Figura No. 22. Interfaz ubicación | 79 |
| Figura No. 23. Interfaz de inicio de sesión..... | 80 |
| Figura No. 24. Panel administrador | 80 |
| Figura No. 25. Panel de control servidor Web | 81 |
| Figura No. 26. Panel de importación de la base de datos | 82 |
| Figura No. 27. Estructura de la base de datos MYSQL | 82 |
| Figura No. 28. Interfaz de datos Fuente: Obando 2016..... | 83 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla No. 1. Comparación Metodologías Ágiles y Tradicionales..... | 31 |
| Tabla No. 2. Roles | 48 |
| Tabla No. 3. Ranking características de los Frameworks..... | 54 |
| Tabla No. 4. Comparativa XP versus SCRUM..... | 55 |
| Tabla No. 5. Ranking de “agilidad”..... | 56 |
| Tabla No. 6. Personas y roles de proyecto | 58 |
| Tabla No. 7. Historia de usuario 1..... | 58 |
| Tabla No. 8. Historia de usuario 2..... | 59 |
| Tabla No. 9. Historia de usuario 3..... | 59 |
| Tabla No. 10. Historia de usuario 4..... | 60 |
| Tabla No. 11. Historia de usuario 5..... | 60 |
| Tabla No. 12. Historia de usuario 6..... | 60 |
| Tabla No. 13. Historia de usuario 8..... | 61 |
| Tabla No. 14. Historia de usuario 9..... | 61 |
| Tabla No. 15. Historia de usuario 10..... | 61 |
| Tabla No. 16. Historia de usuario 11..... | 62 |
| Tabla No. 17. Historia de usuario 12..... | 62 |
| Tabla No. 18. Historia de usuario 13..... | 62 |
| Tabla No. 19. Historia de usuario 14..... | 63 |
| Tabla No. 20. Historia de usuario 15..... | 63 |
| Tabla No. 21. Historia de usuario 16..... | 63 |
| Tabla No. 22. Historia de usuario 17..... | 64 |
| Tabla No. 23. Historia de usuario 18..... | 64 |
| Tabla No. 24. Historia de usuario 19..... | 64 |
| Tabla No. 25. Historia de usuario 20..... | 65 |
| Tabla No. 26. Pila del producto..... | 65 |
| Tabla No. 27. Pila del Sprint | 66 |
| Tabla No. 28. Campos del formulario Registro voluntario..... | 69 |
| Tabla No. 29. Campos del formulario Registro donaciones..... | 70 |
| Tabla No. 30. Campos del formulario Registro empleados..... | 70 |

| | |
|---|----|
| Tabla No. 31. Campos del formulario Registro proyectos..... | 70 |
| Tabla No. 32. Estructura de la programación. | 72 |
| Tabla No. 33. Codificación Modelo | 72 |
| Tabla No. 34. Vista (view)..... | 73 |
| Tabla No. 35. Controlador (controller)..... | 74 |

RESUMEN

El presente trabajo de titulación propone el desarrollo de un sitio web que permite el registro de voluntarios y donaciones para los proyectos de vinculación social que maneja la Fundación “Construyamos Un Sueño” (FCUS - de la ciudad de Latacunga), provincia de Cotopaxi. Los proyectos fomentan la participación de todos los actores sociales que aportan al desarrollo de los sectores más vulnerables. En el sitio web, la implementación de un módulo de donaciones permitirá a la comunidad ayudar con ingresos económicos para la fundación. El sitio web, ha sido diseñado aplicando la metodología ágil SCRUM, acorde a sus características y necesidades. La tecnología seleccionada en el desarrollo e implementación del sitio web utiliza la arquitectura MVC (Modelo Vista Controlador) combinado con el Framework PHP Codeigniter y la base de datos MYSQL, permite que las especificaciones técnicas, los estándares y el análisis realizado, sirvan como estudio base para futuras investigaciones de proyectos relacionados con la vinculación social.

PALABRAS CLAVE

- **SITIO WEB**
- **FRAMEWORK**
- **METODOLOGIA AGIL SCRUM**
- **LENGUAJE PHP**
- **BASE DE DATOS MYSQL**
- **TIPO DE FRAMEWORK CODEIGNITER**

ABSTRACT

The present job proposes the registration of volunteers on a website for social projects in the “Construyamos un Sueño” foundation. The projects encourage the participation of all members who promote to the development of the most vulnerable sectors in the Cotopaxi province. The implementation of a donations module for the community who can contribute with an income for the foundation through a website which has been designed with the best technology and methodology that engages with the development and implementation itself. It uses the architecture of the design pattern: Mode View Controller combined with Codeigniter Framework and connection to the MYSQL database. It is also followed by the development of agile methodology “Scrum”. The technical specifications were assessed for the compliance of applied standards to the construction of a quality website gathering information for future new projects related to social bonding that will serve as a guide for social aid projects.

KEYWORDS

- **SITIO WEB**
- **FRAMEWORK**
- **METODOLOGIA AGIL SCRUM**
- **LENGUAJE PHP**
- **BASE DE DATOS MYSQL**
- **TIPO DE FRAMEWORK CODEIGNITER**

CAPÍTULO I

1. INTRODUCCIÓN AL PROYECTO

1.1. Antecedentes

La fundación construyamos un sueño es una organización privada sin fines de lucro, fundada en el año 1995, tiene personería jurídica mediante Acuerdo Ministerial de ese entonces el Ministerio de Bienestar Social N° 0035-05 del 29 de abril del 2005, con el objetivo social de planificar, ejecutar y evaluar propuestas de desarrollo humano, fomentando la participación de todos los actores sociales, como eje viabilizador del desarrollo de los sectores urbano y rural de la Provincia de Cotopaxi.

El trabajo que ha hecho la organización es visible muchas de las necesidades que tiene la provincia de Cotopaxi, es por eso que ejecutan proyectos con el objetivo de dar una atención integral y garantizar los derechos que como niños, niñas, adolescentes, personas con discapacidad, y adultos mayores se merecen.

En el año en curso y gracias a su larga trayectoria, se ha enmarcado su labor social a favor de los niños, niñas, adolescentes, personas adultas, mayores y personas con discapacidad; la institución ha asumido nuevos retos de trabajo gracias al financiamiento del MIES para lo cual están ejecutando proyectos en donde brindan atención a alrededor de 1556 personas, sin embargo las necesidades diarias que se presentan en cada uno de los proyectos requiere del recurso económico para poder cubrir aquellas necesidades inherentes a la atención que está brindando el equipo técnico, las mismas que no logran ser suplidas en su totalidad por el financiamiento mencionado por parte del MIES, es por ello que la institución ha organizado diferentes eventos, como parte de la autogestión, para

recaudar recursos económicos con el objetivo de brindar continuidad a la labor social que han emprendido.

Los proyectos sociales que durante el año 2015 están en vigencia son:

- Erradicación del trabajo infantil en el cantón Latacunga, atención a 50 niños, niñas y adolescentes.
- Atención a las personas con discapacidad mediante atención en el hogar y la comunidad en el cantón Latacunga, donde se atienden a 200 personas con discapacidad.
- Acogimiento Familiar, atención a 60 hijos/as de las personas privadas de su libertad de la provincia de Cotopaxi.
- Centros Infantiles del Buen Vivir, generan atención a 1050 niños, niñas de 1 a 3 años de edad distribuidos en 25 Centros Infantiles en el cantón Latacunga.
- Atención a 40 adultos mayores del cantón Saquisilí mediante un Centro Diurno de atención integral.
- Atención a 140 adultos mayores del cantón Latacunga mediante atención domiciliaria.
- Casa de Acogimiento "Un Hogar para Todos" ubicada en el cantón Saquisilí, atención a 12 personas entre niños y adultos mayores en situación de riesgo o vulnerados de sus derechos.

A todo esto se adhiere el recurso humano para voluntariado proyectos y diferentes donativos que los ciudadanos aportan en numerosas ocasiones, es por eso que se ven en la necesidad de implementar un sitio web que sistematice las donaciones y el registro de voluntariado mediante la web, para que los donantes contribuyan con seguridad los voluntarios podrán ayudar a seguir fomentando el desarrollo colectivo de la fundación y que esto se vea reflejado como ayuda y atención a la comunidad.

1.2. Planteamiento y formulación del problema

La Fundación Construyamos Un Sueño se encuentra en constante crecimiento en cuanto a los proyectos de vinculación social, aquí intervienen entidades públicas y privadas para brindar ayuda a la comunidad mediante esta institución por lo que estos proyectos son coordinados por la administración de la fundación y la colaboración del voluntariado.

Actualmente existe un avance tecnológico significativo, esto obliga a las Instituciones adaptarse al constante cambio y así sistematizar procesos o flujos de trabajo con innovadoras aplicaciones que integren varias funcionalidades y tecnologías nuevas, que permitan la facilidad de uso y sustenten las necesidades del usuario.

En la fundación construyamos un sueño no existe un sitio web que brinde información y permita a la comunidad participar en la vinculación social con fundaciones o Instituciones sin ánimo de lucro, tampoco existe un sitio web donde las personas mediante el mismo puedan generar donaciones mediante tarjetas de crédito o transferencias bancarias.

1.3. Justificación e importancia

En la fundación construyamos un sueño se analizan los beneficios del sitio web dentro tres factores: tecnológico, social y económico. Como factor tecnológico podemos apreciar la mejor integración de las funciones del sitio no solo como un informativo de la fundación si no como una herramienta más para ayudar a generar ingresos económicos y aumentar recursos humanos como nuevos voluntarios.

Dentro del factor social podremos apreciar una mejor administración de los proyectos participando con más voluntarios en la vinculación de los mismos, con el objetivo social de planificar, ejecutar y evaluar propuestas de

desarrollo humano, fomentando la participación de todos los actores sociales, como eje viabilizado del desarrollo de los sectores urbano y rural de la Provincia de Cotopaxi.

Por último dentro del factor económico se podrá apreciar una optimización en la utilización de recursos humanos y tecnológicos debido a un ingreso económico más para la fundación que ayudara a planificar, ejecutar y evaluar propuestas de desarrollo humano para los sectores urbano y rural.

La integración de un sitio web con la tecnología adecuada para desarrollar un módulo de donaciones, plantea una investigación para identificar la mejor tecnología que optimice la implementación de mismo, con la ayuda del proceso de ingeniería en software y la aplicación de conocimientos de la carrera en base a los diferentes ejes de formación culminaremos este caso de estudio.

1.4. Objetivos generales y específicos

1.4.1. Objetivo General

“Desarrollar un Sitio web para la gestión de donaciones y el registro de voluntarios para la Fundación Construyamos un Sueño, ubicada en la ciudad de Latacunga“

1.4.2. Objetivos Específicos

- Determinar un marco teórico, relacionado al proceso de Ingeniería para la implantación de un sitio Web.
- Conocer los procesos y flujos de trabajo relacionados con las donaciones y voluntariado para la Fundación Construyamos Un Sueño.

- Desarrollar un sitio web basado en la información y procesos identificados de la Fundación Construyamos Un Sueño.
- Aplicar la propuesta en la Fundación Construyamos Un Sueño a través la implementación.

1.5. Metas

Desarrollo de un sitio web dinámico bajo el patrón de diseño Modelo Vista Controlador con la tecnología PHP encargado del registro de voluntarios y donaciones para el apoyo de la Fundación Construyamos Un Sueño, ubicada en la ciudad de Latacunga.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Introducción del capítulo

En el presente capítulo se muestran todos los aspectos teóricos relacionados con la construcción de un sitio web, enfocando el desarrollo a una investigación para la selección de la mejor práctica en metodologías de desarrollo ágil y también la mejor tecnología, relacionada a los frameworks que trabajan bajo el patrón de diseño MVC (Model, View, Controller), desarrollados en lenguaje PHP (Hypertext Pre-Processor), conjuntamente con la utilización de varias herramientas web, las mismas que van a satisfacer las necesidades de navegabilidad del sitio, estimando, el tiempo y esfuerzo para la implementación con varios elementos de configuración del software, podremos guiar este tema de estudio para su correcta implementación.

2.2. Sistemas basados en técnicas WEB.

Los sistemas basados en técnicas web, son una modalidad actual o de mayor aplicación partir de la nueva era de la tecnología WEB, ciertamente esta emplea un tipo de información donde se concentran, dentro de los servidores, todas las tareas y almacenamiento de datos brindando servicios a todos los clientes, pueden ser implementados en redes de área local LAN o redes de áreas extensas WAN o a través de Internet siendo rápidos y dinámicos para el usuario. Actualmente el internet es una tecnología muy necesaria y útil para el desarrollo global, es la fuente de información de todas empresas por lo que está disponible en todo momento existen varios tipos de sistemas de información web citamos algunos de ellos a continuación:

- Sitios web de información
- Sitios web orientados a transacciones
- Sitios web orientados a servicios
- Sitios web acceso a una base de datos
- Sitios web como almacén de datos
- Sitios web de redes Sociales

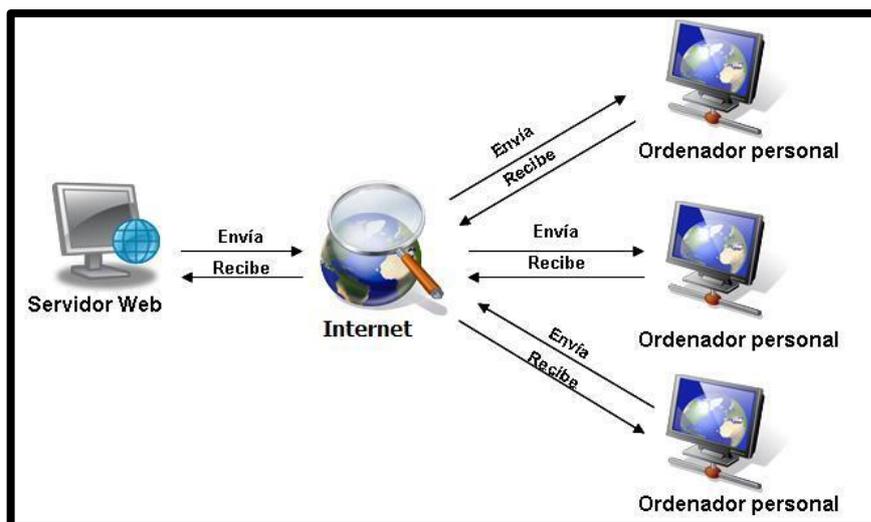


Figura No. 1. Arquitectura de conexión e un sitio web

Fuente: (Bonilla Orozco, 2014)

2.2.1. Diferencia entre un sitio y una página web.

Muchas veces utilizamos los términos sitio web y página web indistintamente. Pero esto es un gran error, dado que una página web pertenece a un sitio web. Un sitio web se compone de una o más páginas web, es un conjunto de ellas. Cada documento o página web contiene texto y gráficos que aparecen como información digital en la pantalla de un ordenador. Un sitio puede contener una combinación de gráficos, texto, audio, vídeo, y otros materiales dinámicos o estáticos. (Paginas web, 2014)

Cada sitio web tiene una página de inicio (en inglés Home Page), que es el primer documento que ve el usuario cuando entra en el sitio web poniendo el nombre del dominio de dicho sitio en un navegador. Normalmente tiene otras

páginas web adicionales. Son gestionados por un administrador, una compañía o una organización. Como medio de comunicación, los sitios web son similares a las revistas, crean y manipulan imágenes digitales y texto, la diferencia principal entre un sitio web y los medios tradicionales es que un sitio web está en una red de ordenadores (Internet) y está codificado de manera que permite que los usuarios interactúen con él. Una vez en un sitio web, puedes realizar compras, búsquedas, enviar mensajes, y otras actividades interactivas. El conjunto de archivos llamados páginas web conforman un sitio web, donde la unión de estos scripts visualiza los contenidos en el internet.

2.3. Definición de ingeniería web

Según (Maigua Soliz, 2009) la ingeniería web es un proceso utilizado para crear, implantar y mantener aplicaciones y sistemas de alta calidad. La ingeniería Web está relacionada con el establecimiento y utilización de principios científicos de ingeniería y gestión, con enfoques sistemáticos y disciplinados del éxito y desarrollo, empleo y mantenimiento de sistemas y aplicaciones Web de alta calidad. Las aplicaciones web se integran cada vez más en las estrategias de negocios de las organizaciones grandes y pequeñas, es necesario que estas aplicaciones sean prácticas, confiables y adaptables. La ingeniería web posibilita estructuras de datos enriquecidas que se utilizan para:

- *Facilitar la búsqueda de contenidos y servicios.*
- *Crear agentes inteligentes para la automatización de procesos.*
- *Extraer, procesar y almacenar conocimiento*
- *Integrar sistemas heterogéneos.*
- *Componer sistemas complejos.*
- *Filtrar información*

El desarrollo de aplicaciones web incorpora métodos de proceso especializados, métodos de ingeniería de software adaptados a características de desarrollo de las aplicaciones web y un conjunto de

importantes tecnologías que permitan un correcto desarrollo de las mismas. Los procesos, métodos y tecnologías (herramientas) proporcionan un enfoque en estratos de la web que es conceptualmente idéntico a los estratos de la ingeniería de software.”

2.3.1. Proceso de la ingeniería web.

Las páginas web con el pasar de los años han venido incrementándose con notoriedad cada día hay más empresas que desean exponer su información en la web como tiendas virtuales, sistemas integrados empresariales, redes sociales todos estos sitios web, son visualizados en todo tipo de dispositivos que por lo menos soporten un navegador web, estos van desde computadores hasta dispositivos móviles como celulares o tablets es por eso que hay la necesidad de realizar sistemas de calidad, que tengan inmediatez navegabilidad, seguridad, evolución continua, estética, mediante un proceso de ingeniería que nos permita diseñar una solución viable para obtener los resultados necesarios para la mejor navegabilidad.

Según el autor (Valdivieso, 2007): “Características como inmediatez, evolución y crecimiento continuos, nos llevan a un proceso incremental y evolutivo, que permite que el usuario se involucre, facilitando el desarrollo de productos que se ajustan mucho lo que éste busca y necesita. Existen algunas actividades importantes que forman parte el proceso, estas son:

- *Formulación*
- *Planificación análisis*
- *Modelización*
- *Generación de páginas*
- *Test y evaluación del cliente”.*

2.3.2. Formulación

En esta etapa se definen las actividades involucradas para el marco de trabajo el cual interactúa con las actividades para el

desarrollo del sitio web, estos pueden modificarse, eliminarse o extenderse en base a las características del problema, el producto, el proyecto y la gente en el equipo. Antes de definir un marco de trabajo se debe reconocer que:

- *Los sitios Web con frecuencia se entregan de manera incremental.*
- *Los cambios ocurrirán frecuentemente.*
- *Los plazos son cortos. (Mirsha, 2013)*

Las personas involucradas en esta etapa son quienes encaminan el proyecto sus participantes son los desarrolladores web, el dueño del producto, y varios integrantes técnicos que nos ayudaran a tomar una correcta decisión en esta etapa de formulación y planeación del proyecto.

De acuerdo con (Pressman, 2011), los actores son todas aquellas personas que colaboraran en el desarrollo exitoso del sistema o aplicación web. Estos actores son todos los ingenieros web que participan en el proyecto y que se pueden organizar en gran medida en los equipos de software tradicionales.

2.3.3. Planificación

Se destaca y se define la planificación de programas, riesgos, y controles del desarrollo se validan cotizaciones, y se entiende el ámbito de trabajo, la confiabilidad de las dimensiones y estimaciones entre tiempo y costo, en esta etapa también se comprende el grado del proyecto definiendo un enfoque para realizar la gestión de cambio.

En la etapa de planeación intensificamos la comunicación con el cliente ya que es vital para darle inicio a la implementación de los requisitos levantados para definir las actividades que se van realizar en esta etapa.

2.3.4. Modelado

Se compone de dos secuencias paralelas de tareas. Una consiste en el diseño y producción del contenido que forma parte de la aplicación. La otra, en el diseño de la arquitectura, navegación e interfaz de usuario. Es conveniente resaltar la importancia del diseño de la interfaz. Independientemente del valor del contenido y servicios prestados, una buena interfaz mejora la percepción que el usuario tiene de éstos. (Cruz Catacora, 2012)

Los escenarios se identifican al inicio del levantamiento de requisitos, son las actividades que tiene cada uno de los actores representados en diferentes formatos o diagramas, pueden ser casos de uso, textos, diagramas de actividad, diagramas de flujo, diagramas de control, diagramas de estado etc, toda esta información nos sirve para realizar un buen modelado, y mediante esto poder completar un buen análisis del negocio.

2.3.5. Generación de páginas

Se integra el contenido, arquitectura, navegación e interfaz para crear estática o dinámicamente el aspecto más visible de la aplicación, las páginas son creadas cumpliendo los requisitos levantados y planificados en las anteriores etapas. (Cruz Catacora, 2012)

En la investigación de María Jesús Lamarca Lapuente se menciona que en el diseño de un sitio web, podemos distinguir dos aspectos:

a. Diseño de la interfaz

Permite guiar al usuario por medio de un sistema visual e informativo adecuado. Para ello hay que disponer la interacción con metáforas, imágenes y conceptos que puedan transmitir significados a través de la pantalla del ordenador. Se debe elegir un modelo y conservar la integridad, uniformidad y coherencia de ese diseño a lo largo de todo el

hiperdocumento. El diseño de la interfaz gráfica engloba conceptos y nociones provenientes de diferentes campos y disciplinas como el diseño gráfico, la informática, los sistemas audiovisuales, la psicología cognitiva, ergonomía, etc.

b. Diseño de las páginas

Disposición de los elementos dentro de las páginas para ser vistos en pantalla, esquemas de contenido, tipografía, rotulado, títulos, disposición de las imágenes y del contenido multimedia, equilibrio entre el contenido textual y gráfico, y la sensación visual, etc. (Lamarca Lapuente, 2013)

2.3.6. Test y evaluación

"El Test busca errores a todos los niveles: contenido, funcional, navegacional, rendimiento, etc. El hecho de que las aplicaciones residan en la red, y que inter-operen en plataformas muy distintas, hace que el proceso de test sea especialmente difícil. El resultado final es sometido a la evaluación del cliente y aceptación por el mismo. Se inicia estableciendo el perfil de los usuarios del sistema y llevando a cabo el análisis conceptual de las tareas, la definición de las restricciones y necesidades de la plataforma de uso y de los principios de diseño a aplicar. Todo ello da lugar a una serie de objetivos de usabilidad que el sistema debe cumplir y que, junto a las guías de estilo, conforman el conjunto estable de requisitos a considerar en el proceso de desarrollo." (Cruz Catacora, 2012)

2.3.7. Entrega y retroalimentación

Con este se refina los requisitos del software que se desarrollará. La interacción ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor los que se deba hacer.

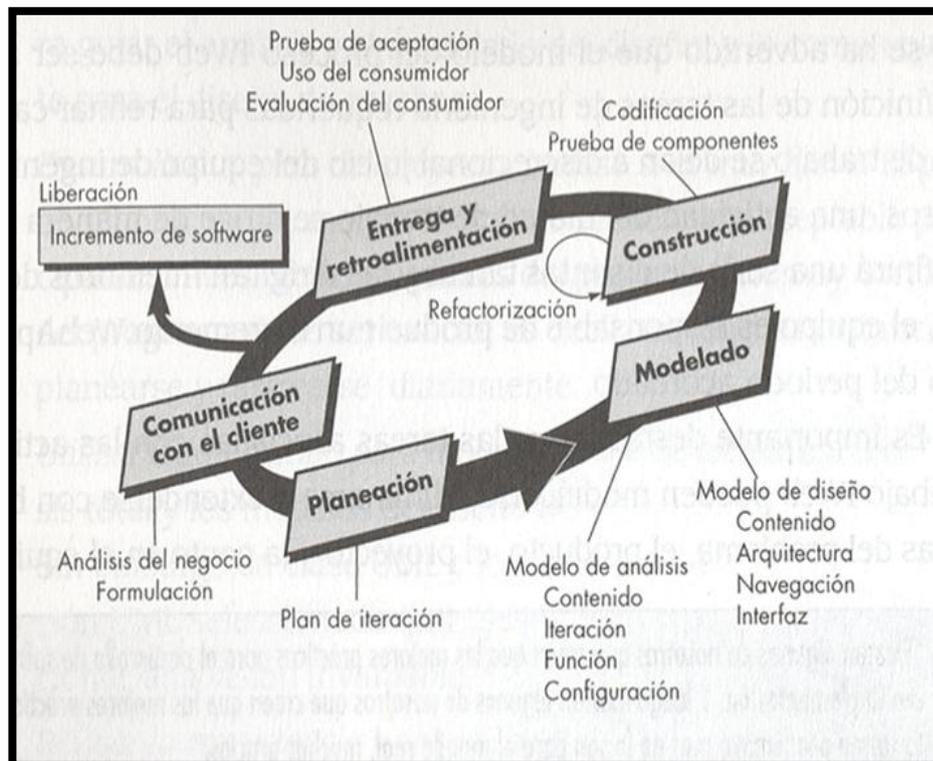


Figura No. 2. Proceso de Ingeniería web

Fuente: (Pressman, 2011)

2.4. Lenguajes de programación web.

Existen numerosos tipos de lenguajes de programación web muchos de ellos tienen características específicas según el uso o la afinidad al momento de desarrollar a continuación se citan los lenguajes que se utilizarán en el desarrollo del presente proyecto ya que posee las características necesarias para su implementación:

- ASP
- HTML
- PHP
- JAVASCRIPT
- JSP

2.4.1. Lenguaje ASP o ASP.net

Es una tecnología del lado de servidor desarrollada por Microsoft para la implementación de sitios web dinámicos. ASP (Active Server Pages) fue liberado en 1996.

“ASP es una tecnología dinámica funcionando del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del script son ejecutadas para enviar al navegador únicamente el código HTML resultante. La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que éste se encuentra únicamente en los archivos del servidor que al ser solicitado a través de la web, es ejecutado, por lo que el usuario no tiene acceso más que a la página resultante en su navegador.” (Taranto Vera & Baquerizo Palma, 2003)

Es un lenguaje que se encuentra del lado del servidor, estas páginas dinámicas son utilizadas para que realicen tareas como conectarse a la base de datos o realizar transacciones como cálculos matemáticos.

Según el autor, (Pérez, 2015), las ventajas y desventajas de este lenguaje son las siguientes.

Ventajas del ASP:

- Admite la programación con Visual Basic Script lo que facilita su implementación.
- Soporta el lenguaje Java Script de Microsoft.
- El lenguaje ASP.NET soporta programación C#, VB.NET y J#. Es un lenguaje completamente orientado a objetos.
- Se comunica de forma impecable con las bases de datos SQL Server.
- ASP.NET ha incrementado notablemente la velocidad y seguridad de los sistemas frente a su hermano mayor ASP.

Desventajas del ASP:

- Es un código propietario, Microsoft desarrolla y comercializa ASP.
- No es fácil de leer e interpretar, se necesita escribir más código para hacer lo mismo que con otros lenguajes como el PHP.
- Muchos hostings y alojamientos web no lo soportan por su alto coste. ASP.NET necesita tener instalado IIS con el Framework .Net.
- El consumo de recursos de ASP.NET es importante por lo que se requieren servidores de mayor capacidad.

2.4.2. HTML

Desarrollado por el World Wide Web. HTML es un lenguaje de etiquetas (también llamado lenguaje de marcado) y las páginas web habituales están formadas por cientos o miles de pares de etiquetas. De hecho, las letras "ML" de la sigla HTML significan "markup language", que es como se denominan en inglés a los lenguajes de marcado. La principal ventaja de los lenguajes de etiquetas es que son muy sencillos de leer y escribir por parte de las personas y de los sistemas electrónicos. La principal desventaja es que pueden aumentar mucho el tamaño del documento, por lo que en general se utilizan etiquetas con nombres muy cortos. Fuente: (Libros web, 2007)

Este lenguaje de programación nos facilita el desarrollo ya que se encuentra del lado del usuario proporcionando velocidad en el sitio ya que sus etiquetas se encuentran en el browser según el autor, (Pérez Valdés, 2007), las ventajas y desventajas de este lenguaje son las siguientes:

a. Ventajas

- Sencillo que permite describir hipertexto.
- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web.

- Archivos pequeños.
- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los exploradores.

b. Desventajas

- Lenguaje estático.
- La interpretación de cada navegador puede ser diferente.
- Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.
- El diseño es más lento.
- Las etiquetas son muy limitadas.

2.4.3. PHP

Significa: Hypertext Pre-processor Es un lenguaje que puede ser embebido en HTML. Gran parte de su sintaxis se basa en lenguaje C, Java, y Perl con características específicas propias de PHP. (Lenguaje de programación y lenguajes de etiqueta, 2014)

El objetivo del lenguaje es permitir a los desarrolladores web escribir con rapidez páginas generadas dinámicamente. Se utiliza para generar páginas web dinámicas, cuyo contenido no es el mismo siempre, puede variar de acuerdo a los nuevos requerimientos de una base de datos, búsquedas o aportaciones de los usuarios, etc.

a. Ventajas

- Lenguaje multiplataforma.
- Fácil de aprender.
- Orientado para desarrollar aplicaciones web donde la información esté en una base de datos.

- Lenguaje modular.
- Mucha documentación debido a su gran popularidad y una gran comunidad. (Web Oficial php.net).
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objetos, clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

b. Desventajas

- Se necesita instalar un servidor web.
- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que aumenten las solicitudes.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- Dificulta la modularización.
- Dificulta la organización por capas de la aplicación.

2.4.4. Lenguaje Javascript

Lenguaje de programación open source interactivo con HTML orientado a objetos, permite mejorar la gestión cliente servidor, dándole efectos dinámicos a la página.

Según el autor (Duarte, 2012):

“JavaScript no tiene nada que ver con el lenguaje de programación Java. JavaScript es un lenguaje de scripting basado en el navegador que ejecuta el código del lado del cliente. Esto significa que cualquier código que se escribe en JavaScript se entrega desde el servidor junto a las páginas web, y todo el código se ejecuta desde el navegador del usuario (en el dispositivo del usuario) en lugar de hacerlo directamente en el servidor donde se encuentra la página web. “

a. Ventajas

- Documentación disponible en la web.
- Es un lenguaje del lado del usuario.
- Tiempo de descarga rápido.
- Soportado por varios navegadores

b. Desventajas

- Es necesario que los usuarios mantengan las librerías de JavaScript, caso contrario perderían la funcionalidad de la página.
- No tiene opciones en 3d.
- No hay seguridad, el código está visible al usuario.

2.4.5. Lenguaje JSP

JSP (Java Server Pages) es un lenguaje de programación web que permite la generación de sitios web dinámicos en lenguaje Java. Es un lenguaje soportado por múltiples plataformas tal como lo hacen PHP y ASP, se ejecutan en del lado del servidor.

Este lenguaje fue desarrollado por Sun Microsystems, empresa que fue absorbida por Oracle. Tiene ventajas similares a las de ASP.NET y es un lenguaje pensado para desarrollar aplicaciones web complejas y potentes. Las páginas se compilan cuando son solicitadas por primera vez.

El código JSP puede integrarse con el código HTML y la extensión de los archivos de este tipo es .jsp.

Posee un motor de páginas basado en los servlets de Java. Para su funcionamiento se necesita tener instalado un servidor Tomcat.

Ventajas del JSP

- El código está bien estructurado y resulta fácil de leer si se conoce bien el lenguaje.
- La parte de JSP dinámico se escribe en Java, permitiendo una integración total con módulos Java y la utilización de un motor de páginas basado en servlets de Java.

Desventajas del JSP:

- *Es un lenguaje bastante complejo y que requiere mayor tiempo de aprendizaje.*
- *El alojamiento web requiere tener instalado un servidor Tomcat. (Pérez, 2015)*

2.5. Patrón de diseño Modelo, vista, controlador (MVC).

En líneas generales, MVC es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores, Model, Views & Controllers. En este caso estudiaremos con detalle estos conceptos, así como las ventajas de ponerlos en marcha cuando desarrollamos.

MVC es un modelo que tiene varias décadas y fue presentado incluso antes de la aparición de la Web. No obstante, en los últimos años ha ganado mucha fuerza y seguidores gracias a la aparición de numerosos frameworks de desarrollo web que utilizan el patrón MVC como modelo para la arquitectura de las aplicaciones web.

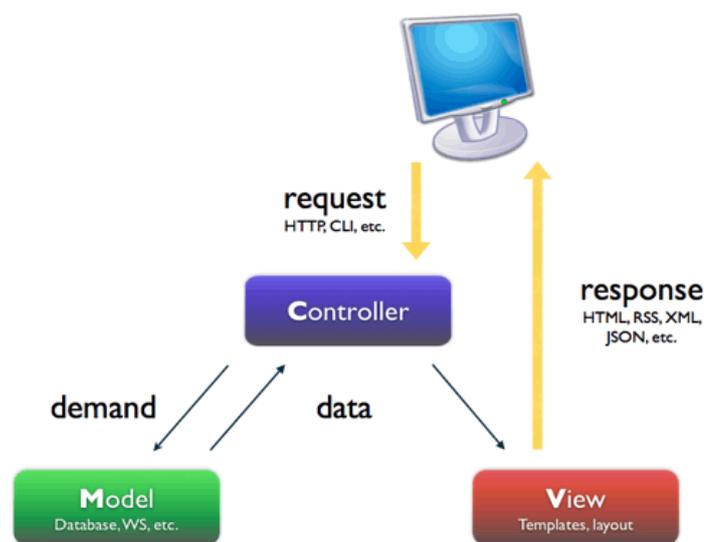


Figura No. 3. Arquitectura del Modelo, Vista, Controlador.

Fuente:_(Libros web, 2014)

El patrón de diseño "modelo vista controlador", es un modelo de desarrollo seguro ya que trabaja en diferentes capas.

2.5.1. Modelo

Es la capa donde se trabaja con los datos, por tanto contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas y harán las correspondientes transacciones.

No obstante, cabe mencionar que cuando se trabaja con MCV lo habitual también es utilizar otras librerías como PDO (PHP Data Objects) o algún ORM (Mapeo Objeto-Relacional) como Doctrine, que nos permiten trabajar con abstracción de bases de datos y persistencia en objetos. Por ello, en vez de usar directamente sentencias SQL, que suelen depender del motor de base de datos con el que se esté trabajando, se utiliza un dialecto de acceso a datos basado en clases y objetos.

- Contiene el núcleo de la funcionalidad (dominio) de la aplicación.
- Encapsula el estado de la aplicación.
- Es independiente del controlador y la vista.

2.5.2. Vista

Las vistas, contienen el código de nuestra aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que nos permitirá direccionar los estados de nuestra aplicación en HTML. En las vistas nada más tenemos los códigos HTML y PHP que nos permite mostrar la salida.

En la vista generalmente trabajamos con los datos, sin embargo, no se realiza un acceso directo a éstos. Las vistas requerirán los datos a los modelos y ellas se generarán la salida, tal como nuestra aplicación requiera.

- Es la presentación del Modelo.
- Puede acceder al Modelo pero nunca cambiar su estado.
- Puede ser notificada cuando hay un cambio de estado en el Modelo.

2.5.3. Controlador

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información, etc.

En realidad es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo. (Álvarez, 2014)

2.5.4. Ventajas y desventajas del Patrón Modelo Vista – Controlador

La principal ventaja de integrar el modelo, la vista y el controlador es que se comunican de manera estable y coherente con las iteraciones que el usuario realiza. Como es lógico, la comunicación entre la vista y el controlador es bastante básica pues están diseñados para operar juntos, otra ventaja es que se puede dividir la lógica de negocio del diseño, facilitará el uso de URL amigables, importantes para el SEO (Search Engine Marketing Professional Organization), la mayoría de frameworks MVC lo controlan. La mayor frecuencia de utilización de este framework por parte de los desarrolladores web es porque se puede diseñar y organizar fácilmente el código en aplicaciones pequeñas y grandes.

a. Ventajas

- Clara separación entre interfaz, lógica de negocio y de presentación, que además provoca parte de las ventajas siguientes.
- Facilidad para la realización de pruebas unitarias de los componentes, así como de aplicar desarrollo guiado por pruebas.
- Reutilización de los componentes.
- Simplicidad en el mantenimiento de los sistemas.
- Se conocerá para desarrollar prototipos rápidos.
- Los desarrollos suelen ser más escalables.

b. Desventajas

- Tener que ceñirse a una estructura predefinida, lo que a veces puede incrementar la complejidad del sistema. Hay problemas que son más difíciles de resolver respetando el patrón MVC.
- La curva de aprendizaje para los nuevos desarrolladores se estima mayor que la de modelos más simples como Webforms.
- La distribución de componentes obliga a crear y mantener un mayor número de ficheros. (tuartdecors.blogspot.com, 2015)

2.6. Framework

Siendo muy simple, es un esquema un esqueleto, o patrón para el desarrollo o la implementación de una aplicación. Al usar este tipo de patrón se separa:

- Gestión de los datos: es la información o los datos que se receipta y envía por medio de la presentación o vista.
- Operaciones: Es la parte lógica de las funcionalidades de navegación del sitio web aquí se programan las operaciones o funciones del sistema.

- Presentación: Esta es la interfaz de usuario en donde se aplicaran estrategias no sólo de diseño sino de funcionalidad.

Los frameworks pueden ser una herramienta muy útil para ahorrar tiempo y ayudar a realizar de una manera dinámica el desarrollo web entre otras herramientas, para así ayudar a acoplar los diferentes componentes de un proyecto.

2.6.1. Características principales.

- El programador no necesita plantearse una estructura global de la aplicación, sino que el framework proporciona un esqueleto que hay que “rellenar”.
- Facilita la colaboración.
- Permite involucrarse con el código fuente de otro programador o incluso con el propio, pasado algún tiempo sabrá lo difícil que es entenderlo y modificarlo; por tanto, todo lo que sea definir y estandarizar va a ahorrar tiempo atendiendo a los desarrollos colaborativos.
- Es más fácil encontrar herramientas (utilidades, librerías) adaptadas al framework concreto para facilitar el desarrollo.

2.6.2. Symfony

Según (González, 2013): “Este Framework PHP es útil para acelerar la creación y el mantenimiento de sus aplicaciones web. Hace uso de las normas existentes de PHP. Proporciona un conjunto de elementos prefabricados que se pueden integrar rápidamente en su aplicación, combinada con una metodología clara para ayudarle a trabajar de forma eficiente y eficaz en las tareas más complejas.”

Ventajas:

- Symfony ofrece una alternativa al enfoque de PHP a través de dos clases que permiten interactuar la petición HTTP y la respuesta de una manera más fácil.
- Fácil de instalar y configurar en la mayoría de plataformas.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.

Desventajas:

- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.

2.7. Tipos de Framework WEB

Existen varios tipos algunos sirven para diseñar mejor las plantillas, otros nos ayudan al desarrollo y en algunos casos para la animación. El desarrollo de software es un camino largo y complejo, el cual requiere mucho tiempo. Sin embargo, la utilización de un framework puede ayudarle a desarrollar los proyectos con más rapidez mediante la reutilización de componentes y módulos genéricos, y trabajar mejor sobre una base estructural unificada. El uso de un framework también facilita la escalabilidad y el mantenimiento de su código.

Según (González, 2013) los 6 mejores Frameworks PHP, considerados no solo por sus características, sino por la tendencia de mayor uso, las cuales serán de gran ayuda al momento de programar, al eliminar el tiempo y el esfuerzo necesarios para desarrollar componentes genéricos, puede dedicar más tiempo a las tareas y funciones específicas, y centrarse en el código de alta calidad, se comparten a continuación.

2.7.1. Codeigniter

Este framework posee un conjunto de herramientas para la creación de una web con características precargadas y visualización web, Codeigniter viene con una documentación clara. Algunas otras características interesantes de este framework PHP es que incluye muy poca configuración, sin librerías monolíticas de gran escala, compatibilidad con alojamiento estándar, sin reglas de codificación restrictivas, sin necesidad de lenguaje de plantillas y muchos más.

Ventajas:

- Las páginas se procesan más rápido, el núcleo de Codeigniter es bastante ligero.
- Es sencillo de instalar.
- Reutilización de código, desarrollo ágil.
- Existe abundante documentación en la red.
- Facilidad para crear nuevos módulos, páginas o funcionalidades.
- Acceso a librerías públicas y clases.
- Estandarización del código. Fundamental cuando hay que tocar código hecho por otra persona o cuando trabaja más de una persona en un mismo proyecto.
- Separación de la lógica y arquitectura de la web, el MVC.

Desventajas:

- Codeigniter es mucho menos rígido que otros frameworks.
- Define una manera de trabajar en constante cambio y actualizaciones por lo que hay que revisar cada una de las nuevas versiones.

2.7.2. Yii Framework

Considerado como un framework PHP rápido, estable, seguro y de alto rendimiento, Equipado con el patrón Modelo-Vista-Controlador (MVC), esquema de almacenamiento en caché con múltiples características, acceso y autenticación basado en roles, objetos de acceso a bases de datos (DAO), widgets habilitados para Ajax y documentos detallados; Yii sirve como un framework ideal para el desarrollo de aplicaciones web empresariales, aplicaciones de redes sociales y mucho más.

Ventajas:

- Acelerar el proceso de desarrollo.
- Reutilizar código a lo largo de varios proyectos.
- Estabilidad.
- Separación de capas lógicas (Modelo, vista, controlador).
- Seguridad, sobre todo frente vulnerabilidad SQL

Desventaja:

- Poca información y documentación de colaboradores.

2.8. Bases de datos

Una base de datos es una herramienta para recopilar y organizar información. En las bases de datos, se puede almacenar información sobre

personas, productos, pedidos, etc. Muchas bases de datos empiezan siendo una lista en un programa de procesamiento de texto o en una hoja de cálculo. En la base de datos el conjunto de datos relacionados se encuentra agrupado y estructurado para brindar información a los usuarios de los sistemas informáticos. Las principales características de los sistemas de base de datos, son:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.
- Existen diferentes tipos de bases de datos, por cuanto para este estudio práctico en particular; se tomará en cuenta bases de datos orientadas a la web.

2.8.1. Tipos de bases de Datos

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, o la utilidad de la misma. Según la variabilidad de los datos almacenados los tipos de bases de datos que se acoplan a la tecnología **PHP** las cuales se describe a continuación:

a. MySQL

Es un sistema de administración de bases de datos relacionales (Database Management System). MySQL es una aplicación que nos permite gestionar archivos de bases de datos. Al funcionar como base de datos relacional, funciona con múltiples tablas para almacenar y organizar la

información. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

Ventajas:

- MySQL es Open Source.
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación. Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet.
- El software MySQL tiene la licencia GPL2.

Desventajas:

- No existe compatibilidad con software comercial.
- Un gran porcentaje de las utilidades de MySQL no están documentadas (La Info, 2013).

b. PostgreSQL

Postgres, desarrollada originalmente en el Departamento de Ciencias de la Computación de la Universidad de California en Berkeley, fue pionera en muchos de los conceptos de bases de datos relacionales orientadas a objetos que ahora empiezan a

*estar disponibles en algunas bases de datos comerciales.
(Cárdenas, 2015)*

Ventajas

- Ampliamente popular - Ideal para tecnologías Web.
- Fácil de Administrar.
- Su sintaxis SQL es estándar y fácil de aprender.
- Multiplataforma.
- Capacidades de replicación de datos.
- Soporte empresarial disponible.

Desventajas

- *Sin experticia, configurar llega a ser un caos.*
- *Es fácil de vulnerar sin protección adecuada.*
- *El toolset (set de herramientas) empresarial tiene un costo adicional por suscripción anual.*
- *Realizar revisiones llegar a ser una labor manual y tediosa para el DBA.*
- *Reducida cantidad de tipos de datos. (Kinderman, Silva, & Velásquez, PostGreSQL, 2011)*

2.9. Metodologías ágiles y tradicionales de desarrollo web.

Existen muchas empresas que están optando por la utilización de varias propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo hay dos tipos, tradicionales y ágiles, estas poseen cualidades únicas que implican trabajo en equipo, permitiendo ahorro de tiempo en el proceso de desarrollo, por lo tanto el cliente podrá visualizar e interactuar con los desarrolladores para optimizar las características del software, en las metodologías tradicionales hay un mayor énfasis en la documentación con el objetivo de obtener un software más eficiente, mientras que las metodologías ágiles se centran en la capacidad de dar respuestas rápidas

ante el desarrollo interactivo con el cliente ya que mantiene pocos artefactos y esto la hace más adaptable al cambio.

A continuación se presenta un cuadro comparativo de las metodologías ágiles versus las tradicionales, poniendo énfasis en las características más importantes.

Tabla No. 1.

Comparación Metodologías Ágiles y Tradicionales.

| Metodología Ágil | Metodología Tradicional |
|---|---|
| <ul style="list-style-type: none"> • Pocos Artefactos. • Pocos Roles, más genéricos y flexibles • No existe un contrato tradicional debe ser bastante flexible. • Cliente es parte del equipo de desarrollo. • La arquitectura se va definiendo y mejorando a lo largo del proyecto • Basadas en heurísticas provenientes de prácticas de producción de código • Se esperan cambios durante el proyecto. | <ul style="list-style-type: none"> • Más Artefactos. El modelado es esencial, mantenimiento de modelos. • Más Roles, más específicos. • Existe un contrato prefijado. • El cliente interactúa con el equipo de desarrollo mediante reuniones. • Se promueve que la arquitectura se defina tempranamente en el proyecto. • Énfasis en la definición del proceso: roles, actividades y artefactos • Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo |

Fuente: (Letelier & Penadés, 2006)

2.9.1. Metodologías tradicionales

De acuerdo con el autor (Acuña, 2010) “La filosofía de desarrollo de las metodologías, aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales o Pesadas.

Estas metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en

la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar”.

Entre las metodologías tradicionales o pesadas podemos citar:

- RUP (Rational Unified Procces)
- MSF (Microsoft Solution Framework)
- Win-Win Spiral Model
- Iconix

No detallaremos cada uno de estos tipos ya que nuestro caso de estudio se enfoca en metodologías de desarrollo ágil.

2.9.2. Metodologías ágiles

Su filosofía es centrarse en el factor humano o el producto software. Estas metodologías le dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Las metodologías ágiles están revolucionando la manera de producir software, de ésta manera en el presente trabajo se presentan algunos tipos de metodologías ágiles para el desarrollo web.

a. Metodología XP

Según: (Letelier & Penadés, 2006)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de

software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y dando un buen clima de trabajo. XP se basa en la comunicación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

a1. Características de la metodología XP

Las características fundamentales de la metodología XP son las siguientes: (Ing. Software (Equipo 02), 2012)

- **Desarrollo iterativo e incremental:** pequeñas mejoras del producto, unas tras otras, en cada entrega del producto según la planificación se hace pruebas por el usuario.
- **Programación en parejas:** Se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera el código es revisado y discutido mientras se escribe es más importante que la posible pérdida de productividad inmediata.
- **Código compartido:** en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- **Simplicidad en el código:** es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

a2. Proceso XP

Según (Guerrero, 2008), el ciclo de vida ideal de XP consiste de cuatro fases:

Primera Fase: Planificación del proyecto.

En esta primera fase se debe hacer primero una recopilación de todos los requerimientos del proyecto, también debe haber una interrelación con el usuario, y se debe planificar bien entre los desarrolladores del proyecto que es lo que se quiere para el proyecto para así lograr los objetivos finales.

Segunda Fase: Diseño

Se sugiere que hay que conseguir diseños simples y sencillos. Para procurar hacerlo todo lo menos complicado posible para el usuario o cliente, para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo para desarrollarlo. En esta fase se logrará crear parte del proyecto la parte física (lo bonito) la interfaz que tendrá el usuario o cliente con el proyecto.

Tercera Fase: Codificación

Como se indicó anteriormente, el cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de X.P. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada. En esta fase de la codificación los clientes y los desarrolladores del proyecto deben estar en comunicación para que los

desarrolladores puedan codificar todo lo necesario para el proyecto que se requiere, en esta fase está incluido todo lo de codificación o programación por parte de los desarrolladores del proyecto.

Cuarta Fase: Pruebas

Uno de los pilares de la metodología X.P es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando.

Para esta fase lo que se implementa es el uso de test que son pruebas que se le hacen al proyecto o como ya se dijo a los códigos que se vayan implementando.

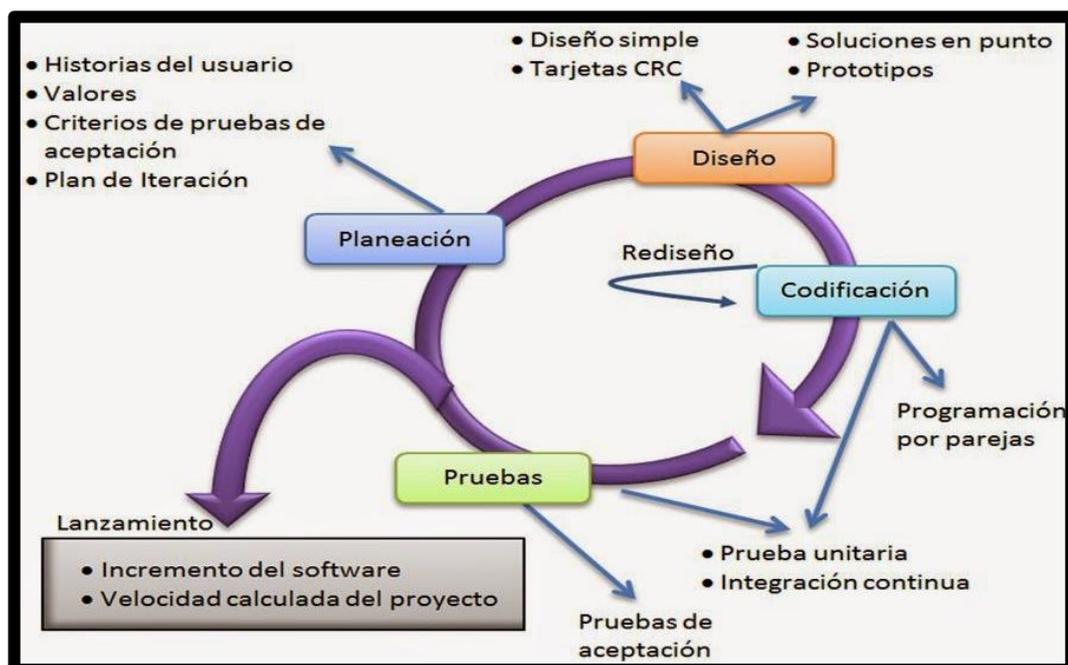


Figura No. 4. Fases XP

Fuente: (Slideplayer, 2015)

a3. Roles XP

Describiremos los actores o roles de acuerdo con el autor, (Beck, 1999) , como: "Una explicación de la programación extrema.

Beck. Cada rol tiene diferentes responsabilidades y tareas durante todo el proceso de desarrollo.

Programador

El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.

Cliente

El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.

Encargado de pruebas (Tester)

El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

Encargado de seguimiento (Tracker)

El encargado de seguimiento proporciona retroalimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones. También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina

cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

Entrenador (Coach)

Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

- Experto en XP responsable del proceso en su conjunto
- Identifica las desviaciones y reclama atención sobre las mismas
- Guía al grupo de forma indirecta (sin dañar su seguridad ni confianza)
- Interviene directamente si es necesario
- Comprender rápidamente el problema

Consultor

Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

Gestor (Big boss)

Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

a4. Artefactos XP

Existen varios artefactos como historias de usuario, tarjetas de ingeniería casos de prueba de aceptación, tarjetas C.R.C., a continuación mostraremos su objetivo: (Tripod, 2015)

Historias de usuario

El primer paso de cualquier proyecto que siga la metodología X.P es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas.

Casos de prueba de aceptación

Son tarjetas que se elaboran para realizar las pruebas de cada historia de usuario.

Tarea de ingeniería

Son tarjetas que se elaboran para ayudar y simplificar la programación de una historia de usuario.

Tarjetas CRC

Describen las clases utilizadas en la programación de una historia.

Ventajas

Una de las ventajas de la programación extrema es que se adapta al desarrollo de sistemas pequeños y grandes; optimiza el tiempo de desarrollo; permite realizar el desarrollo del sistema en parejas para complementar los conocimientos; el código es sencillo y entendible, además de la poca documentación a elaborar para el desarrollo del sistema.

Desventajas

No tiene la definición del costo y el tiempo de desarrollo; el sistema va creciendo después de cada entrega al cliente y nadie puede decir que el cliente no querrá una función más; se necesita de la presencia constante del usuario, lo cual en la realidad es muy difícil de lograr.

Otra desventaja es la programación en parejas, algunos desarrolladores son celosos del código que escriben y no les es grato que alguien más modifique las funciones que realizó o que su código sea desechado por no cubrir el estándar. (Dirección de Comunicación Universitaria Departamento de Prensa, 2012)

b. Metodología Ágil Scrum.

En todos los proyectos de software siempre existirá la necesidad del cambio en todo su proceso de desarrollo y etapas de planificación por lo que se creó esta metodología de desarrollo ágil que facilita su realización.

Según (Scrum Alliance, 2016):

Scrum es un framework de desarrollo ágil de software. El trabajo es estructurado en ciclos de trabajo llamados sprint, iteraciones de trabajo con una duración típica de dos a cuatro semanas. Durante cada sprint, los equipos eligen de una lista de requerimientos de cliente priorizados, llamados historias de usuarios, para que las características que sean desarrolladas

primero sean las de mayor valor para el cliente. Al final de cada sprint, se entrega un producto potencialmente comercial.

b1. El proceso

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos, iteraciones de un mes natural y hasta de dos semanas, si así se necesita. Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite, en la Figura No. 5 podemos apreciar el proceso de desarrollo.

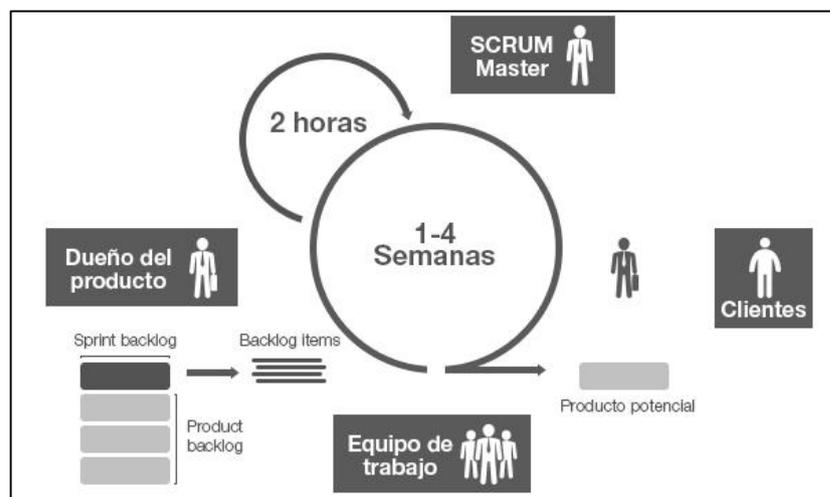


Figura No. 5. Desarrollo Scrum

Fuente: (Scrum manager, 2013)

*Los procesos que se llevan a cabo en Scrum son los siguientes:
(Scrum manager, 2013)*

Planificación de la iteración

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

- **Selección de requisitos:** El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.
- **Planificación de la iteración:** El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las tareas.
- **Ejecución de la iteración.** Cada día el equipo realiza una reunión de sincronización (15 minutos máximo). Cada miembro del equipo inspecciona el trabajo que los demás participantes realizan, dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo, para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido. En la reunión cada miembro del equipo responde a tres preguntas:
 - ¿Qué he hecho desde la última reunión de sincronización?
 - ¿Qué voy a hacer a partir de este momento?
 - ¿Qué impedimentos tengo o voy a tener?

Inspección y adaptación

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

- ***Demostración:*** *El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, re-planificando el proyecto.*

- **Retrospectiva:** El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. (Scrum manager, 2013)

b2. Artefactos

Existen varios artefactos en los que Scrum se apoya para desarrollar su metodología estos son los siguientes:

Pila del producto: (product backlog) lista de requisitos de usuario, que a partir de la visión inicial del producto crece y evoluciona durante el desarrollo. Esta pila del producto es una lista ágil de los requisitos del cliente o requisitos del sistema:

- **Simple:** Expresados de forma breve, con una sentencia para cada uno, habitualmente con formato de historia de usuario o similar.
- **Estimados:** Está estimado el esfuerzo de construcción de cada requisito.
- **Priorizados:** Ordenados según la importancia para el cliente o responsable de la lista.

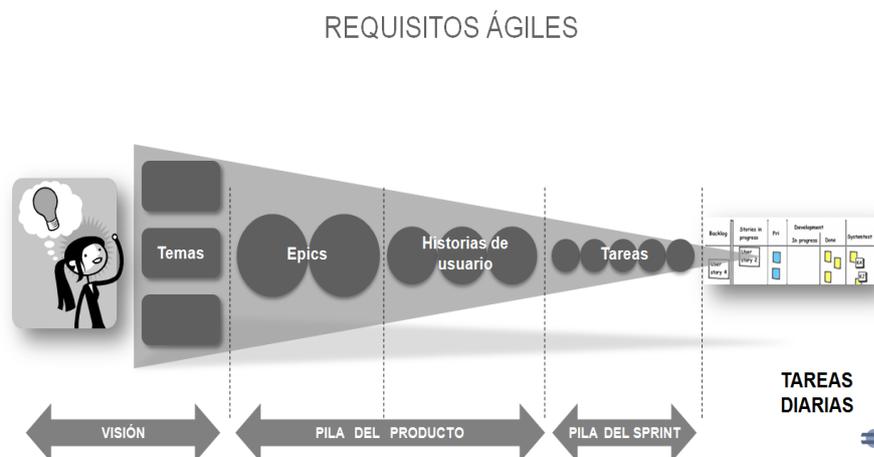


Figura No. 6. Pila del producto y pila del sprint

Fuente: (Scrum manager, 2014)

La ingeniería del software clásica diferencia dos ámbitos de requisitos:

- Requisitos del sistema
- Requisitos del software

Los requisitos del sistema forman parte del proceso de adquisición, y por tanto es responsabilidad del cliente la definición del problema y de las funcionalidades que debe aportar la solución.

No importa si se trata de gestión tradicional o ágil. La pila del producto es responsabilidad del cliente, aunque se aborda de forma diferente en cada caso.

Formato de la pila del producto

Scrum prefiere la comunicación verbal o de visualización directa, a la escrita. La pila del producto no es un documento de requisitos, sino una herramienta de referencia para el equipo. Si se emplea formato de lista o historia de usuario, es recomendable que al menos incluya la siguiente información para cada elemento:

- Identificador único de la funcionalidad o trabajo.
- Descripción de la funcionalidad/requisito, denominado “historia de usuario”.
- Campo o sistema de priorización.
- Estimación del esfuerzo necesario.

En la definición del proyecto podemos dividir en varios objetivos expresados como historias de usuario que en algunas ocasiones poseen un formato dependiendo del tipo de proyecto, funcionamiento del equipo y la organización, pueden ser aconsejables otros campos:

- Observaciones.
- Criterio de validación.
- Persona asignada.
- N° de Sprint en el que se realiza.
- Módulo del sistema al que pertenece.
- Entre otros.

| Historia de Usuario | |
|--|-------------------------------|
| Número: 1 | Usuario: Cliente |
| Nombre historia: Cambiar dirección de envío | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Baja |
| Puntos estimados: 2 | Iteración asignada: 1 |
| Programador responsable: José Pérez | |
| Descripción: Quiero cambiar la dirección de envío de un pedido. | |
| Validación: El cliente puede cambiar la dirección de entrega de cualquiera de los pedidos que tiene pendientes de envío. | |

Figura No. 7. Historia de usuario Scrum

Fuente: (Scrum manager, 2015)

Todas las historias de usuario se priorizan de acuerdo al valor que le da el cliente y se realiza con una descripción breve del objetivo final.

He aquí un sencillo ejemplo de pila de producto: (Srum manager, 2014)

| Id | Prioridad | Descripción | Est. | Por |
|----|-----------|---|------|-----|
| 1 | Muy alta | Plataforma tecnológica | 30 | AR |
| 2 | Muy Alta | Interfaz de usuario | 40 | LM |
| 3 | Muy Alta | Un usuario se registra en el sistema | 40 | LM |
| 4 | Alta | El operador define el flujo y textos de un expediente | 60 | AR |
| 5 | Alta | xxx | 999 | CC |

Figura No. 8 Pila de producto

Fuente: (Smanager, 2013)

Pila del sprint: Es una lista de tareas que va a realizar el equipo en una iteración, para construir un incremento. Para cada una registra la información:

- Descripción breve.
- Persona que la tiene asignada.
- Esfuerzo pendiente para terminarla.
- Información

Es útil porque descompone el proyecto en unidades de tamaño adecuado para determinar el avance a diario e identificar riesgos y problemas sin necesidad de procesos complejos de gestión. Es también una herramienta de soporte para la comunicación directa del equipo.

Condiciones

- Realizada de forma conjunta por todos los miembros del equipo.
- Cubre todas las tareas identificadas por el equipo para conseguir el objetivo del sprint.
- Sólo el equipo la puede modificar durante el sprint.
- Las tareas demasiado grandes deben descomponerse en otras más detalladas. Se deben considerar “grandes” las tareas que necesitan más de un día para realizarse.
- Es visible para todo el equipo. Idealmente en un tablero o pared en el mismo espacio físico donde trabaja el equipo.

Formato y soporte

Son soportes habituales:

- Tablero físico o pared.
- Hoja de cálculo.

- Herramienta colaborativa o de gestión de proyectos.

Y sobre el más adecuado a las características del proyecto, oficina y equipo, lo apropiado es diseñar el formato más cómodo para todos, teniendo en cuenta los siguientes criterios:

- Incluir la siguiente información: Pila del sprint, persona responsable de cada tarea, estado en el que se encuentra y tiempo de
- Trabajo que queda para completarla. (Srum manager, 2014)

| SPRINT | | | INICIO | DURACIÓN | | | | | | | | | | | | | | |
|----------------------------|-----------|-----------|----------|----------|----------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|----|----|
| 1 | | | 1-mar-07 | 12 | J | V | L | M | X | J | V | L | M | J | V | L | | |
| | | | | | 1-mar | 2-mar | 5-mar | 6-mar | 7-mar | 8-mar | 9-mar | 12-mar | 13-mar | 15-mar | 16-mar | 19-mar | | |
| | | | | | 23 | 23 | 19 | 16 | 16 | 13 | 9 | 9 | 9 | 9 | 9 | 9 | | |
| | | | | | 276 | 246 | 216 | 190 | 178 | 158 | 110 | 110 | 110 | 110 | 110 | 110 | | |
| SPRINT BACKLOG | | | | | ESFUERZO | | | | | | | | | | | | | |
| Tarea | Estad | Responsal | | | | | | | | | | | | | | | | |
| Descripción de la tarea 1 | Terminada | Luis | 16 | 16 | 16 | 16 | 16 | 16 | | | | | | | | | | |
| Descripción de la tarea 2 | Terminada | Luis | 12 | 8 | | | | | | | | | | | | | | |
| Descripción de la tarea 3 | Terminada | Luis | 4 | 4 | 4 | 4 | 4 | | | | | | | | | | | |
| Descripción de la tarea 4 | Terminada | Elena | 8 | 4 | | | | | | | | | | | | | | |
| Descripción de la tarea 5 | Terminada | Elena | 16 | 16 | 4 | | | | | | | | | | | | | |
| Descripción de la tarea 6 | Terminada | Elena | 6 | 6 | 2 | | | | | | | | | | | | | |
| Descripción de la tarea 7 | Terminada | Antonio | 16 | 4 | | | | | | | | | | | | | | |
| Descripción de la tarea 8 | Terminada | Antonio | 16 | 16 | 20 | 12 | 4 | | | | | | | | | | | |
| Descripción de la tarea 9 | Terminada | Antonio | 12 | 2 | | | | | | | | | | | | | | |
| Descripción de la tarea 10 | En curso | Luis | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |
| Descripción de la tarea 11 | Pendiente | Luis | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| Descripción de la tarea 12 | Terminada | Luis | 14 | 14 | 14 | 14 | 14 | 14 | 14 | | | | | | | | | |
| Descripción de la tarea 13 | En curso | Antonio | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 6 | | | | | | | | |

Figura No. 9. Ejemplo pila sprint

Fuente: (Smanager, 2013)

Sprint

Nombre que recibe cada iteración de desarrollo, es el núcleo central que genera el pulso de avance por tiempos prefijados, cada sprint produce un incremento. Este ciclo de tiempo en el que se desarrolla cada incremento iterativo del producto. En función de las características del proyecto y el criterio del equipo, lo habitual es realizar sprints de duración no inferior a una semana ni mayor de un mes.



Figura No. 10. Modelo evolutivo iterativo

Fuente: (Smanager, 2013)

Las primeras descripciones del ciclo de Scrum (Schwaber & Beedle) establecían una duración mínima de dos semanas y máxima de dos meses. Tanto entonces como ahora, los ciclos de desarrollo con iteraciones de más de dos meses no se consideran ágiles. (Scrum manager, 2013)

Incremento

Resultado de cada sprint, El incremento es la parte de producto producida en un sprint y tiene como característica el estar completamente terminada y operativa, en condiciones de ser entregada al cliente. No se debe considerar como incremento a prototipos, módulos o sub-módulos, ni partes pendientes de pruebas o integración.

- Cada elemento de la pila del producto se refiere a funcionalidades entregables, no a trabajos internos del tipo “diseño de la base de datos”.
- Se produce un “incremento” en cada iteración.

Sin embargo es una excepción frecuente el primer sprint. En el que objetivos del tipo “contrastar la plataforma y el diseño” pueden resultar

necesarios, e implican trabajos de diseño o desarrollo de prototipos para contrastar las expectativas de la plataforma o tecnología que se va a emplear. Teniendo en cuenta esta excepción habitual:

Incremento es la parte de producto realizada en un sprint potencialmente entregable: terminada y probada.

Si el proyecto o el sistema requiere documentación, o procesos de validación y verificación documentados, o con niveles de independencia que implican procesos con terceros, éstos también tienen que estar realizados para considerar que el incremento está “hecho”

b3. Roles Scrum

En Scrum, el equipo se focaliza en construir software de calidad. La gestión de un proyecto Scrum se centra en definir cuáles son las características que debe tener el producto a construir qué construir, qué no y en qué orden y en vencer cualquier obstáculo que pudiera frenar la tarea del equipo de desarrollo ,el equipo Scrum está formado por los siguientes roles:

Tabla No. 2.

Roles

| Comprometidos | Implicados |
|---|--|
| Propietario del producto Miembros del equipo | Otros interesados (dirección, gerencias, comerciales, marketing, etc.) |

Fuente: (Scrum manager, 2015)

Scrum master: Persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el Product Owner para maximizar el ROI.

Product owner: Representante de la empresa, accionistas y clientes que usan el software. Se focaliza en la parte de negocio y él es responsable del proyecto para entregar un valor superior al dinero invertido.

Traslada la visión del proyecto al equipo, formaliza las prestaciones en historias a incorporar en el Product Backlog y las re-prioriza de forma regular.

Team: Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a cabo las historias a las que se comprometen al inicio de cada sprint (Softeng, 2015).

CAPÍTULO III

3. SELECCIÓN DE LA TECNOLOGÍA

En el presente capítulo se realizará una revisión, análisis, comparación de todo el marco teórico mencionado anteriormente para la correcta selección de la mejor tecnología que utilizaremos para el proceso de desarrollo y su programación, tomando como punto de partida aspectos importantes como los requisitos de usuario, el tiempo de desarrollo e implementación, seleccionando el mejor framework que deberá acoplarse a las necesidades del usuario y los tiempos de entrega del proyecto aplicando el esquema de trabajo que sugiere la metodología ágil, para que todos estos aspectos que son pilares importantes para el cumplimiento de nuestra investigación, lleven a cabo un buen desarrollo sistemático y un mayor control del desarrollo del Sitio web.

3.1. Selección del lenguaje de Programación.

Utilizaremos PHP como lenguaje principal ya que existe mucha documentación, tutoriales, ejemplos de código, foros, etc. El rendimiento de PHP es muy bueno y verdaderamente eficiente en cuanto al tiempo de respuesta del sitio web, utilizando un servidor básico se puede atender varias peticiones al día, sus características técnicas son necesarias para este proyecto es un lenguaje fácil de aprender y nos orienta para desarrollar aplicaciones web donde la información esté en una base de datos robusta, posee una buena integración con la mayoría de conectores a base de datos. MySQL, Postgre SQL, Oracle, etc. Su programación orientada a objetos hace que nuestro proyecto sea dinámico en cuanto a los tiempos de programación y los requerimientos técnicos este lenguaje es de código libre y gratuito con bibliotecas muy amplias con funciones nativas existen múltiples Frameworks PHP que permiten que nuestro código sea más ordenado, estructurado, manejable y permiten trabajar utilizando patrones de

diseño cómo Modelo-Vista-Controlador que es lo que buscamos en nuestro proyecto.

3.2. Selección Base de Datos

Las bases de datos son sumamente utilizadas para el registro de la información. Es una colección de información organizada de tal manera que un programa de computadora puede rápidamente seleccionar datos deseados por lo que una buena plataforma. Para nuestro caso de estudio es MySQL ya que esta se acopla con php y es una base de datos muy robusta que maneja niveles de seguridad, es de código abierto y no posee costosas licencias por soporte.

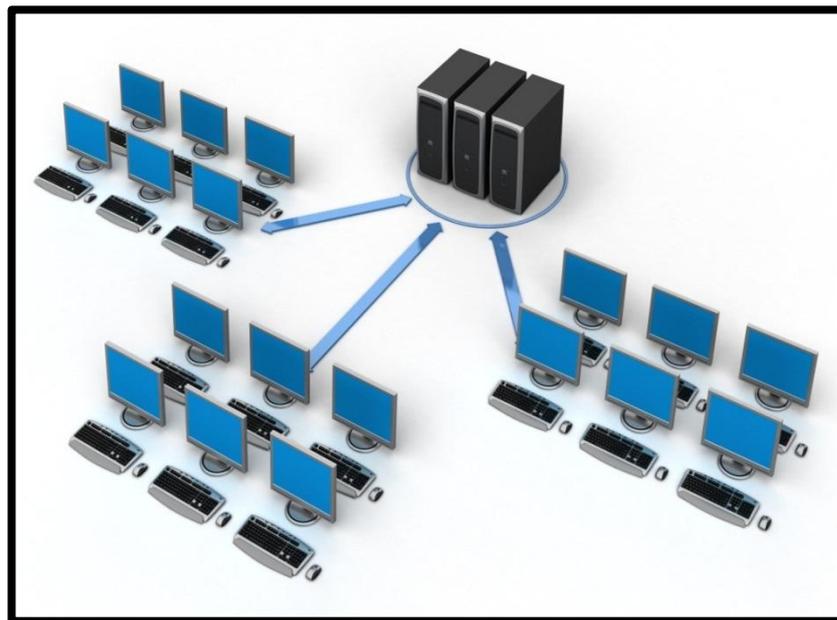


Figura No. 11. Bases de datos locales

Fuente (Webs Uvigo, 2015)

3.2.1. MySQL

Elegimos MySQL como motor de base de datos de la aplicación por su fácil configuración con el lenguaje de programación PHP y su integración en distintos sistemas operativos, esta base de datos es Open Source (Código

abierto), su velocidad al realizar las operaciones, es lo que le hace uno de los gestores con mejor rendimiento para la comunicación de datos en nuestro sitio web, posee un bajo costo en requerimientos para la elaboración de la base de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos, MySQL usa la licencia GPL para evitar altos costos en nuestro proyecto' .

3.3. Frameworks

Para la selección de nuestro framework estudiamos tres tipos, con los que hicimos una comparación técnica para utilizar la mejor tecnología para nuestro proyecto entre Yii, Codeigniter, Symfony, todos estos con la misma arquitectura de diseño que manejan tecnologías de desarrollo web **PHP** que trabaja en 3 capas MVC (Modelo vista controlador); este patrón de diseño nos facilitará el desarrollo incremental del sitio web ahorrándonos tiempo de codificación y facilitándonos un desarrollo dinámico.

3.3.1. Comparativa de frameworks.

Los frameworks elegidos para la realización de la comparación están enfocados a la programación orientada a objetos donde mantienen una estructura definida para acelerar los tiempos de desarrollo con otras librerías y un breve mantenimiento y escalabilidad en el sistema. Los requisitos técnicos que vamos a emplear, son indicadores comunes para cada uno de los frameworks, estas características principales que poseen, permiten que cada uno se acople a las necesidades del desarrollo del sitio web, serán nuestro ranking de comparación, la principal característica es la forma dinámica y versátil al programar mejorando los tiempos de entrega al usuario, también se toma en cuenta la seguridad en el envío y manejo de información algo muy importante que necesita nuestro proyecto es que sea adaptable a librerías y plugins de desarrollo web. Para nuestra investigación tomaremos en cuenta los indicadores que mencionamos anteriormente

conjuntamente con otras características serán nuestra guía para la correcta elección.

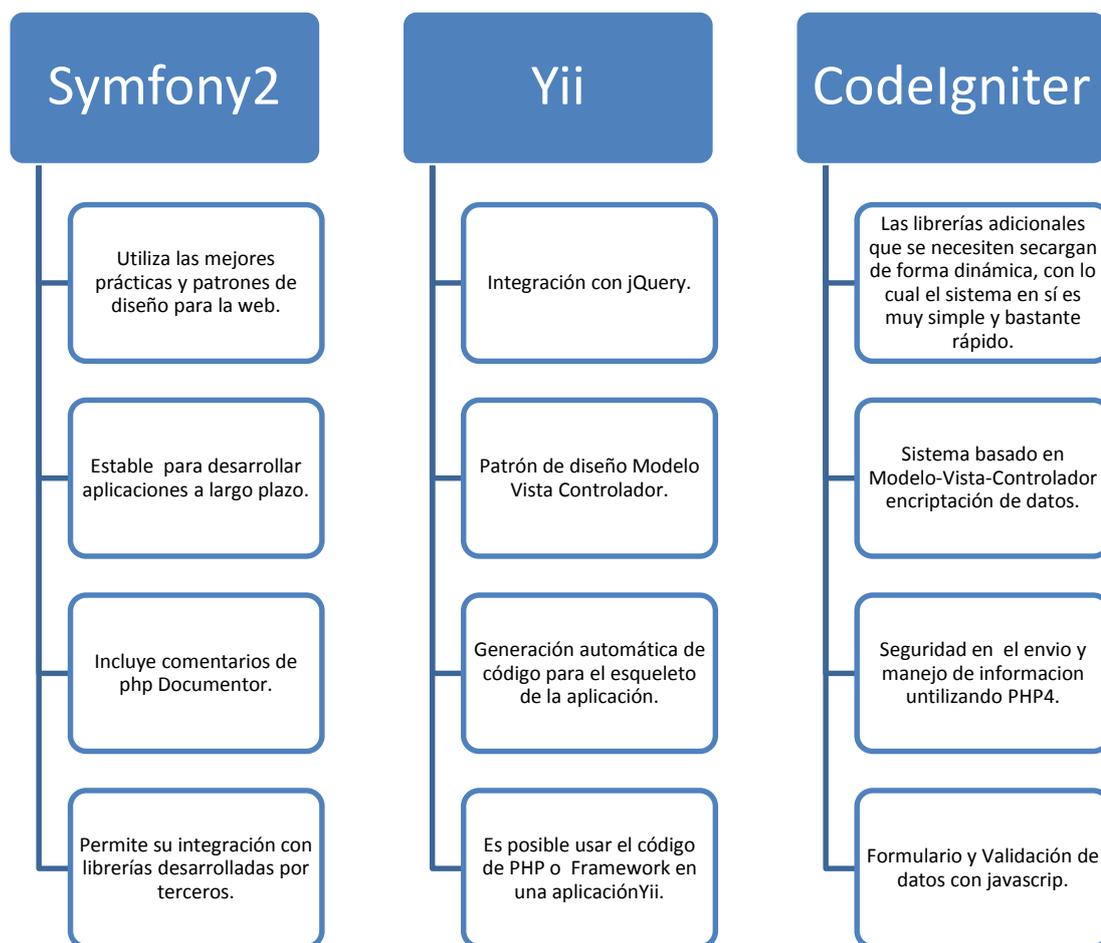


Figura No. 12. Características de Framework MVC

En base al estudio del marco teórico y la realización de la comparación entre las mejores tecnologías podremos definir a las siguientes características como indicadores de las necesidades del usuario para poder escoger a la mejor tecnología en desempeño de acuerdo al MVC (Modelo, vista, controlador), todos estos ítems son los requisitos tecnológicos fundamentales y necesarios para la correcta aplicación del desarrollo de nuestro proyecto.

Tabla No. 3.

Ranking características de los Frameworks.

| Características de los Frameworks | Symfony2 | Yii | CodeIgniter |
|--|-----------|-----------|-------------|
| Acelerar el proceso de desarrollo. | 4 | 4 | 5 |
| Reutilizar código ya existente. | 3 | 3 | 4 |
| Promover buenas prácticas de desarrollo. | 3 | 4 | 5 |
| Uso de patrones. | 5 | 4 | 5 |
| Permitir la utilización de bibliotecas alterna | 4 | 4 | 5 |
| Soporte para PHP. | 4 | 4 | 4 |
| Desarrollo de software multiplataforma. | 4 | 4 | 5 |
| Portabilidad entre plataformas. | 3 | 4 | 5 |
| Componentes modulares con Plugins. | 2 | 3 | 5 |
| Total: | 32 | 34 | 43 |

El framework que recibe mayor puntaje en base a sus características principales, es CodeIgniter y el que recibió menor puntaje fue Symfony2.

Selección

De acuerdo a estas características técnicas la mejor tecnología para nuestro sitio es “**CodeIgniter**” ya que utiliza un patrón de diseño que se acopla al desarrollo y el tiempo de ejecución del proyecto siguiendo como principal característica la forma dinámica y versátil al programar, esta cuenta con una alta seguridad en el manejo de la información gracias a su programación en capas y al patrón de diseño (MVC) Modelo, vista, controlador, su compatibilidad con varias librerías de diseño hacen que sea la mejor opción para nuestro sitio web.

3.4. Metodologías de desarrollo ágil.

Las metodologías de desarrollo ágil están aplicadas a la toma de decisiones, basados en el desarrollo iterativo e incremental para así crear un software sin errores centrándose en el control del proceso de desarrollo y enfocándose en un mejor nivel de satisfacción del usuario o cliente respondiendo a los cambios continuos que puedan surgir a lo largo del proyecto. Una de las más importantes metodologías con ciclos rápidos e interacciones capaces de permitir cambios de último momento son XP y

Scrum, son muy adaptables al cambio de requisitos y esto facilitará la creación de un sitio web para un proceso de implementación rápido y dinámico.

3.4.1 Comparativa de las metodologías.

El objetivo de la investigación es enfocarnos en la metodología de desarrollo ágil, que cubra las necesidades de nuestro sitio web tomando en cuenta los tiempos cortos y el cambio constante que necesitamos para culminar el proyecto en el que desarrollamos partiendo de una comparativa de metodologías escogidas anteriormente para que, en esta investigación, se tenga en cuenta varias características se realiza la comparación de en base a las características propias de la metodología ágil estas son:

Tabla No. 4.

Comparativa XP versus SCRUM

| XP Extreme Programming | SCRUM |
|--|--|
| <ul style="list-style-type: none"> • Es una metodología de desarrollo de software Ágil enfocado en el código. • La programación se la realiza en parejas. • La duración del proyecto es de 1 - 3 semanas de acuerdo a este estándar. • El código original van manteniéndolo simple, para ir mejorando continuamente con los cambios • Hay continuas pruebas automatizadas al código. • Simplicidad del código. | <ul style="list-style-type: none"> • Es enfocado para la gestión del desarrollo de software iterativo. • Adaptabilidad a posibles cambios • La Historias de usuario son implementadas en un tiempo de 2 - 4 semanas como máximo. • Alineamiento entre el cliente y el equipo de desarrollo. • Cada miembro de Scrum trabaja de forma individual. • Retroalimentación mediante los sprints o periodos del proyecto. |

Siguiendo las necesidades del usuario frente a las características que nos ofrece las metodologías de desarrollo podemos fijarnos que Scrum trabaja en el plazo requerido para el desarrollo de cada sprint o iteración para el levantamiento de historias de usuario y los requisitos del sistema, el cual realiza una retroalimentación cada fin de iteración en el que podremos

adaptar cualquier tipo de cambio o modificación del sistema. Por lo que concluimos que la mejor metodología para el desarrollo del proyecto es Scrum, por el tiempo en cada sprint, y su mínima documentación, adaptabilidad al cambio y perfecto acoplamiento a los requerimientos para el producto final del sitio web.

Tabla No. 5.

Ranking de “agilidad”

| Características Metodología (CM) | Scrum | XP |
|----------------------------------|-------|------|
| Resultados | 5 | 5 |
| Simplicidad | 5 | 5 |
| Adaptabilidad | 5 | 3 |
| Excelencia técnica | 4 | 3 |
| Prácticas | 4 | 5 |
| Sistema web cambiante | 5 | 5 |
| Colaboración | 5 | 3 |
| Total promedio: | 4,57 | 4,14 |

En el análisis detallado anteriormente observamos que el costo referencial de la combinación de tecnologías son calificados sobre 5 puntos, pudimos comparar en un promedio que la metodología de desarrollo de ágil que superó en puntaje es “SCRUM” por lo será la metodología para el desarrollo del proyecto acorde a las necesidades del usuario.

CAPÍTULO IV

4. DESARROLLO DEL SITIO WEB

En este capítulo se tomó en consideración la elección de las mejores tecnologías seleccionadas anteriormente, siendo aplicadas conjuntamente con sus respectivos estándares, plantillas y métricas, para la correcta gestión de la propuesta de desarrollo de un sitio web bajo el patrón de diseño Modelo vista controlador, que permita la gestión del registro de voluntarios y donaciones para dar soporte e información a la Fundación “Construyamos un sueño”, ubicada en la ciudad de Latacunga, todos estos artefactos servirán para evaluar el correcto desempeño en el proceso de desarrollo de software para su implementación final.

Este capítulo se centra en la aplicación de la metodología ágil Scrum ya que cubre las necesidades del usuario final en cuanto a tiempos de entrega y características del producto, para así definir un adecuado seguimiento de nuestro proyecto, con este esquema de trabajo siendo flexible al constante cambio, que es lo que necesitamos para poder detallar un correcto nivel de los requisitos del sitio web, hablaremos acerca del diseño y desarrollo siguiendo con las recomendaciones de la metodología Scrum para la creación de artefactos que nos guiarán en la arquitectura del sitio web.

En el capítulo 3 seleccionamos a Codeigniter como tecnología de desarrollo MVC (Modelo, vista, controlador), este framework PHP, nos ayudará en la creación de módulos dinámicos para la reutilización del código en las clases que posee por defecto, mencionamos las más importantes a continuación, describimos cada una de las capas programadas para nuestro sitio web, así como también el diseño de la base de datos y la interface para nuestro proyecto.

4.1. Utilización de la metodología ágil Scrum en el Proyecto.

Tabla No. 6.

Personas y roles de proyecto

| Persona | Contacto | Rol |
|-------------------|--|-----------------|
| Alejandro Obando | alejo_sof@hotmail.com | Scrum Manager |
| Andrea Bustamante | fcus@yahoo.es | Product owner |
| Alejandro Obando | alejo_sof@hotmail.com | Team Developers |

4.1.1. Pila del producto.

Esta unidad contiene las necesidades a un nivel detallado como para permitir a los diseñadores crear un sistema que satisfaga a los usuarios, el equipo de prueba deberá planificar y realizar las evaluaciones que demuestren si el sistema cumple o no con los requerimientos.

Para la recopilación de información de la pila del producto nos apoyamos en la documentación que maneja la Fundación “Construyamos un Sueño”, con documentos de apoyo, formatos de registros establecidos en sus procesos de trabajo y entrevistas al personal involucrado con los procesos a automatizar. Estos registros se encuentran compilados en el anexo 3.

4.1.2. Historias de usuario

Tabla No. 7.

Historia de usuario 1

HU-001

Nombre de la historia: Crear Barra de menú.

Número: 1

Usuario: **Usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Alta

Alta

Puntos estimados: **40**

Iteración asignada: 1

Programador responsable: **Alejandro Obando**

Descripción: Necesito que la página visualice una barra de menú con varios elementos como: Proyectos, Información de la empresa, Ubicación, Visión, Misión, Teléfonos de Contacto, Correos, etc. El usuario podrá registrar su información para colaborar como voluntario en los proyectos sociales que se realizan en la Fundación, así como registrar varios tipos de donaciones en beneficio de las personas.

CONTINÚA →

HU-001

Validación: El usuario podrá navegar libremente por el sitio utilizando el menú como acceso a las diferentes gestiones que manejará.

Tabla No. 8.**Historia de usuario 2**

HU-002:

Nombre de la historia: Crear el diseño del Home page (página de inicio)

Número: **2**

Usuario: **Usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Alta

Alta

Puntos estimados: **40**

Iteración asignada: **1**

Programador responsable: **Alejandro Obando**

Descripción: La página debe tener un carrusel de imágenes aleatorias con la vinculación a cada proyecto, que brinde al cliente información visual de las actividades que se están realizando en la Fundación y también como un informativo de los beneficios para otras personas, también necesito que se muestre un breve resumen de los proyectos en los que trabajan actualmente de modo que se vincule a la información detallada de cada uno de ellos

Validación: El cliente podrá navegar libremente, ver la información y beneficios de las personas de los proyectos que se están realizando, mediante imágenes y otros vínculos de la página.

Tabla No. 9.**Historia de usuario 3**

HU-003

Nombre de la historia: Crear Footer (pie de página), con módulo PayPal.

Número: **3**

Usuario: **Usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Alta

Alta

Puntos estimados: **40**

Iteración asignada: **1**

Programador responsable: **Alejandro Obando**

Descripción: En el pie de la página del sitio web, agregar un link donde los usuarios puedan realizar donaciones mediante PayPal.

Validación: El usuario podrá realizar donaciones mediante la página siempre y cuando posea una cuenta PayPal o tenga una tarjeta de crédito con los bancos que tienen convenio con PayPal.

Tabla No. 10.**Historia de usuario 4**

HU-004

Nombre de la historia: Poner información de la dirección en el Footer (pie de página)

Número: 4

Usuario: **Usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Alta**Alta**

Puntos estimados: 20

Iteración asignada: 1

Programador responsable: **Alejandro Obando****Descripción:** En el pie de la página del sitio web, quiero que los usuarios puedan tener información de la fundación como su ubicación y teléfonos de contacto.**Validación:** El usuario podrá visualizar en el footer (pie de página) las direcciones y teléfonos de contacto como informativo de la Fundación.**Tabla No. 11.****Historia de usuario 5**

HU-005

Nombre de la historia: Crear informativo en el inicio

Número: 5

Usuario: **Usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Alta**Alta**Puntos estimados: **40**

Iteración asignada: 1

Programador responsable: **Alejandro Obando****Descripción:** En la página inicial, realizar un blog informativo con imágenes aleatorias de nuestros colaboradores dentro de un carrusel y acceso a los proyectos con la comunidad.**Validación:** El cliente podrá ver por medio del blog informativo las actividades más recientes así como también imágenes de los colaboradores.**Tabla No. 12.****Historia de usuario 6****Nombre de la historia: Crear Sub menú ¿Quiénes somos?**

Número: 6

Usuario: **Usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Baja**Baja**Puntos estimados: **20**

Iteración asignada: 1

Programador responsable: **Alejandro Obando****Descripción:** Crear un enlace que permita a las personas ver la información de la fundación: misión, visión, objetivo social y nuestra historia.**Validación:** El usuario podrá ver en el menú Proyectos, por medio del link ¿Quiénes somos? varios paneles componentes para la información de la Fundación: misión, visión, objetivo social y nuestra historia.

Tabla No. 13.**Historia de usuario 8**

HU-008

Nombre de la historia: Crear informativo de Proyectos Sociales

Número: 8

Usuario: **Usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Media**Medio**Puntos estimados: **40**

Iteración asignada: 1

Programador responsable: **Alejandro Obando**

Descripción: En el link de Proyectos sociales quiero visualizar un blog informativo con imágenes de los proyectos de la Fundación, todo esto con una breve descripción de cada imagen y un link de acceso a cada uno de ellos con su respectiva información.

Validación: El cliente podrá ver, por medio el blog, los proyectos más recientes así como también imágenes de los colaboradores en cada uno de ellos.

Tabla No. 14.**Historia de usuario 9**

HU-009

Nombre de la historia: Crear sub menú Proyectos culminados

Número: 9

Usuario: **Usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Baja**Baja**

Puntos estimados:20

Iteración asignada: 1

Programador responsable: **Alejandro Obando**

Descripción: Las personas pueden ver información detallada con imágenes de los proyectos que culminaron e información como actividades y sus beneficiarios.

Validación: El usuario tendrá acceso a la información de actividades y beneficiarios de cada proyecto.

Tabla No. 15.**Historia de usuario 10**

HU-010

Nombre de la historia: Crear sub menús Donaciones

Número: 10

Usuario: **Usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Media**Media**Puntos estimados: **40**

Iteración asignada: 1

Programador responsable: **Alejandro Obando**

Descripción: En la barra de menú necesito crear una pestaña que habilite dos elementos de Donaciones: uno para acceso directo al registro de Donaciones mediante PayPal y otro para el formulario de registro de Donaciones físicas.

Validación: El usuario podrá registrar sus donaciones en la pestaña donaciones del menú ya sean económicas con una cuenta PayPal o físicas por medio del formulario de registro.

Tabla No. 16.**Historia de usuario 11**

HU-011

Nombre de la historia: Crear top banner (Banner superior) de videos de la fundación

Número: 11

Usuario: **Usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Baja**Baja**Puntos estimados: **20**

Iteración asignada: 1

Programador responsable: **Alejandro Obando**

Descripción: En el módulo de donaciones necesito un banner en la parte inferior que muestre varios videos de los proyectos sociales en los que se trabajan actualmente.

Validación: Los usuarios podrán visualizar el trabajo que realiza la fundación por medio de varios videos disponibles en esta página.

Tabla No. 17.**Historia de usuario 12**

HU-012

Nombre de la historia: Crear pestaña Registro de voluntario.

Número: 12

Usuario: **Usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Alta**Alta**Puntos estimados: **40**

Iteración asignada: 1

Programador responsable: **Alejandro Obando**

Descripción: En la pestaña de Registro de voluntario necesito ver un formulario que permita ingresar la información que posteriormente vamos a revisar en la fundación, se presenta un modelo de la ficha de registro de voluntario que se encuentra en el anexo C.

Validación: El usuario podrá registrar los siguientes datos de un voluntario: nombre, apellido, cédula, dirección, ciudad, profesión, móvil, correo, teléfono, interés, motivación.

Tabla No. 18.**Historia de usuario 13**

HU-013

Nombre de la historia: Crear gestión de voluntario.

Número: 13

Usuario: **Usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Alta**Alta**Puntos estimados: **40**

Iteración asignada: 1

Programador responsable: **Alejandro Obando**

Descripción: En la pestaña de Registro de voluntario necesito registrar, eliminar, y editar los siguientes campos nombre, apellido, cédula, dirección, ciudad, profesión, móvil, correo, teléfono, interés, motivación, todos estos registros quiero manipular en un panel de control.

Validación: El usuario podrá registrar los siguientes datos de un voluntario: nombre, apellido, cédula, dirección, ciudad, profesión, móvil, correo, teléfono, interés, motivación.

CONTINÚA →

Tabla No. 19.**Historia de usuario 14**

HU-014

Nombre de la historia: Crear pestaña de contactos con geo-localización.

Número: 14

Usuario: **Usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Alta**Alta**Puntos estimados: **40**

Iteración asignada: 2

Programador responsable: **Alejandro Obando****Descripción:** En la pestaña de contactos se desplegarán dos sub opciones más una de ubicación y otra de login para iniciar sesión.**Validación:** El usuario podrá navegar en los sub menús de contactos.**Tabla No. 20.****Historia de usuario 15**

HU-015

Nombre de la historia: Crear pestaña ubicación

Número: 15

Usuario: **Usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Alta**Alta**Puntos estimados: **40**

Iteración asignada: 2

Programador responsable: **Alejandro Obando****Descripción:** En la pestaña de ubicación quiero visualizar la posición con geo-localización para que el usuario pueda ubicar con el mapa nuestra dirección exacta también debe tener un formulario de registro de sugerencias o contactarnos por correo.**Validación:** El usuario podrá visualizar la dirección con Google Maps y enviar comentarios del sitio.**Tabla No. 21.****Historia de usuario 16**

HU-016

Nombre de la historia: Crear Login para el administrador del sistema.

Número: 16

Usuario: **Administrador**

Prioridad en negocio:

Riesgo en desarrollo:

Alta**Alta**Puntos estimados: **40**

Iteración asignada: 2

Programador responsable: **Alejandro Obando****Descripción:** El administrador del sistema tendrá un acceso en el menú de Contactos para que pueda ingresar al panel de control.**Validación:** El administrador del sistema podrá acceder mediante el Login.

Tabla No. 22.**Historia de usuario 17****HU-017****Nombre de la historia: Crear el panel de control del administrador**

Número: 17

Usuario: **Administrador**

Prioridad en negocio:

Riesgo en desarrollo:

Alta**Alta**Puntos estimados: **40**

Iteración asignada: 2

Programador responsable: **Alejandro Obando**

Descripción: El administrador del sistema contará con un panel de control para gestionar las donaciones físicas y registro de voluntarios pudiendo manipular funcionalidades básicas en cada uno de ellos.

Validación: El administrador del sistema podrá navegar en un panel de control para gestionar todas las funcionalidades del sistema.

Tabla No. 23.**Historia de usuario 18****HU-018****Nombre de la historia: Crear Reporte de registros para voluntarios y donaciones físicas.**

Número: 18

Usuario: **Administrador**

Prioridad en negocio:

Riesgo en desarrollo:

Alta**Alta**Puntos estimados: **40**

Iteración asignada: 2

Programador responsable: **Alejandro Obando**

Descripción: El administrador del sistema podrá manipular la información registrada por parte de los usuarios o voluntarios y los datos de los registros de donaciones físicas para la generación de distintos tipos de reportes necesarios para la gestión de información de la Fundación.

Validación: El administrador del sistema podrá acceder mediante el login de usuario a su panel de control y manejar gestiones de la información registrada.

Tabla No. 24.**Historia de usuario 19****HU-019****Nombre de la historia: Crear Registro de empleado en el panel de administrador.**

Número: 19

Usuario: **Administrador**

Prioridad en negocio:

Riesgo en desarrollo:

Alta**Alta**Puntos estimados: **40**

Iteración asignada: 2

Programador responsable: **Alejandro Obando**

Descripción: El administrador del sitio tendrá acceso a un formulario donde podrá registrar empleados para la administración del sitio, así como podrá gestionar su información.

Validación: El administrador podrá ingresar nuevos empleados para el panel de control manipulando altas bajas y modificaciones de la información.

Tabla No. 25.

Historia de usuario 20

HU-020

Nombre de la historia: Crear Registro de donación

Número: 20

Usuario: **Administrador-usuario**

Prioridad en negocio:

Riesgo en desarrollo:

Alta**Alta**Puntos estimados: **40**

Iteración asignada: 2

Programador responsable: **Alejandro Obando**

Descripción: El administrador del sitio tendrá acceso a un formulario donde podrá manipular los registros de las donaciones pudiendo realizar altas, bajas, y cambios por medio del sitio web indicando que tipo de donación quiere hacer si es económica, física este proceso lo definimos por medio de una gestión que el administrador está habilitado para realizar El administrador mediante el panel de control también podrá registrar donaciones directas al sistema.

Validación: El administrador podrá manipular los registros de Donaciones mediante el panel de control realizando su ingreso de sesión.

4.1.3. Pila del producto

Tabla No. 26.

Pila del producto

| PILA DEL PRODUCTO | | | | |
|-------------------|------------|--|-------------------|----------|
| ID | Categoría | Requerimiento | Estimado en horas | Sprint # |
| HU-001 | Desarrollo | Crear barra de menú. | 40 | 1 |
| HU-002 | Desarrollo | Crear el diseño del Home page (página de inicio) | 20 | 1 |
| HU-003 | Desarrollo | Crear Footer con módulo PayPal. | 20 | 1 |
| HU-004 | Desarrollo | Crear Footer (pie de página), con módulo PayPal. | 20 | 1 |
| HU-005 | Desarrollo | Crear informativo en el inicio. | 40 | 1 |
| HU-006 | Desarrollo | Crear sub menú de Proyectos | 20 | 1 |
| HU-007 | Desarrollo | Crear Sub menú ¿Quiénes somos? | 40 | 1 |
| HU-008 | Desarrollo | Crear informativo de Proyectos Sociales | 20 | 1 |
| HU-009 | Desarrollo | Crear sub menú Proyectos culminados | 40 | 1 |
| HU-010 | Desarrollo | Crear sub menús Donaciones | 40 | 1 |
| HU-011 | Desarrollo | Crear top banner (Banner superior) de videos de la fundación | 20 | 1 |
| HU-012 | Desarrollo | Crear pestaña Registro de voluntario. | 40 | 1 |
| HU-013 | Desarrollo | Crear gestión de voluntario. | 40 | 1 |
| HU-014 | Desarrollo | Crear pestaña de contactos con geo-localización. | 40 | 2 |
| HU-015 | Desarrollo | Crear login para el administrador del sistema. | 40 | 2 |
| HU-016 | Desarrollo | Crear Login para el administrador del sistema. | 40 | 2 |
| HU-017 | Desarrollo | Crear el panel de control del | 40 | 2 |

CONTINÚA →

| PILA DEL PRODUCTO | | | | |
|-------------------|------------|---|-------------------|----------|
| ID | Categoría | Requerimiento | Estimado en horas | Sprint # |
| | | administrador | | |
| HU-018 | Desarrollo | Crear Reporte de registros para voluntarios y donaciones físicas. | 40 | 2 |
| HU-019 | Desarrollo | Crear Registro de empleado en el panel de administrador. | 40 | 2 |
| HU-020 | Desarrollo | Crear Registro de donación | 40 | 2 |

4.1.4. Pila del sprint

Esta lista descompone las funcionalidades de la pila del producto en tareas necesarias para construir un incremento o iteración es decir una parte completa del producto que se puede poner en producción tomando en cuenta que el orden del sprint sea priorizado por el cliente. Fuente: (Álvarez Santos & Gómez Pinto, 2015)

Tabla No. 27.

Pila del Sprint

| PILA DEL SPRINT | | | | | |
|-----------------|------------|--|-------------|-------------------|--------|
| ID | Categoría | Requerimiento | Responsable | Estimado en horas | Estado |
| 1.1 | Desarrollo | Crear Diseño del sitio | A Obando | 10 | |
| 1.2 | Desarrollo | Crear slider de imágenes aleatorias | A Obando | 10 | |
| 1.3 | Desarrollo | Crear informativo de noticias en el inicio | A Obando | 10 | |
| 1.4 | Desarrollo | Realizar un pequeño blog de proyectos | A Obando | 10 | |
| 2.1 | Desarrollo | Crear una barra de menú | A Obando | 40 | |
| 3.1 | Desarrollo | Crear pie de inicio con modulo PayPal | A Obando | 35 | |
| 3.2 | Desarrollo | Crear cuenta PayPal y validar inf. | A Obando | 5 | |
| 4 | Desarrollo | Crear información del pie de inicio | A Obando | 20 | |
| 5 | Desarrollo | Crear informativo en el inicio | A Obando | 20 | |
| 5.1 | Desarrollo | Crear un carrusel de imágenes de colaboradores | A Obando | 20 | |
| 6 | Desarrollo | Crear información misión, visión, historia, objetivo social. | A Obando | 20 | |
| 7 | Desarrollo | Crear blog de proyectos sociales | A Obando | 40 | |
| 8 | Desarrollo | Crear sub menú de proyectos culminados | A Obando | 20 | |
| 9 | Desarrollo | Crear blog informativo | A Obando | 40 | |
| 10 | Desarrollo | Crear menú de donaciones | A Obando | 10 | |
| 10.1 | Desarrollo | Crear menú PayPal | A Obando | 15 | |

CONTINÚA →

| PILA DEL SPRINT | | | | | |
|-----------------|------------|---|-------------|-------------------|--------|
| ID | Categoría | Requerimiento | Responsable | Estimado en horas | Estado |
| 10.2 | Desarrollo | crear módulo de registro de donaciones | A Obando | 15 | |
| 11 | Desarrollo | Crear banner de videos | A Obando | 20 | |
| 12 | Desarrollo | Crear registro de voluntario | A Obando | 40 | |
| 13 | Desarrollo | Crear ubicación con geo localización | A Obando | 20 | |
| 13.1 | Desarrollo | Crear formulario de comentarios. | A Obando | 20 | |
| 14 | Desarrollo | Crear ubicación con geo localización | A Obando | 40 | |
| 15 | Desarrollo | Crear login para administración | A Obando | 40 | |
| 16 | Desarrollo | Crear el panel de control administrador | A Obando | 40 | |
| 17 | Desarrollo | Reporte de voluntarios y donaciones. | A Obando | 40 | |
| 18 | Desarrollo | Registro de empleados | A Obando | 40 | |
| 19 | Desarrollo | Crear registro de donación A,B,C. | A Obando | 40 | |
| 20 | Test | Pruebas de las funcionalidades. | A Obando | 40 | |

4.2. Diseño de los formularios

Para que las personas ingresen sus datos fácilmente, se ha analizado la forma en que se presentarán los formularios, de tal manera que sean comprensibles, organizados, interactivos y fáciles de entender para los usuarios del sistema, sobre todo sea información concreta y necesaria para poder gestionarla, a continuación, explicamos los formularios del sitio web.

4.2.1. Formulario registro de voluntario.

De acuerdo con los requisitos, los campos necesarios en este formulario serán los siguientes.

| Campos Obligatorios | |
|--|--|
| <p>Registro de voluntarios</p> <input type="text" value="Ingrese su Nombres"/> <input type="text" value="Ingrese sus Apellidos"/> <input type="text" value="Ingrese su Empresa"/> <input type="text" value="Ingrese su Email"/> <input type="text" value="Ingrese su Telefono"/> <input type="text" value="Ingrese interes"/> <input type="text" value="motivacion"/> <input type="button" value="Registrar"/> | <ul style="list-style-type: none"> • Nombres • Apellidos • Dirección • Email • Teléfono |

4.2.2. Formulario registro de Donaciones

De acuerdo con los requisitos, los campos necesarios en este formulario serán los siguientes:

| Campos Obligatorios | |
|---|--|
| <p>Registro de donaciones</p> <input type="text" value="Ingrese su Nombres"/> <input type="text" value="Ingrese sus Apellidos"/> <input type="text" value="Ingrese su Empresa"/> <input type="text" value="Ingrese su Telefono"/> <input type="text" value="Ingrese su Email"/> <input type="text" value="Ingrese comentario"/> <input type="text" value="tipo donacion"/> <input type="button" value="Registrar"/> | <ul style="list-style-type: none"> • Nombres • Apellidos • Empresa • Email • Teléfono |

4.2.3. Formulario registro empleados

De acuerdo con los requisitos, los campos necesarios en este formulario serán los siguientes.

| Campos Obligatorios | |
|--|---|
| <p>Registro de Empleado</p> <input type="text" value="Ingrese su Nombres"/> <input type="text" value="Ingrese sus Apellidos"/> <input type="text" value="Ingrese su CI"/> <input type="text" value="Ingrese su Telefono"/> <input type="text" value="Ingrese su Email"/> <input type="text" value="Ingrese clave"/> <input type="text" value="confirme clave"/> <input type="button" value="Registrar"/> | <ul style="list-style-type: none"> • Nombres • Apellidos • Cedula • Email • Password |

4.3. Diseño de la base de datos

Se ha analizado el diseño de base de datos tomando en cuenta los módulos que se van a desarrollar y la información que se maneja en cada una de las gestiones que se contemplan el proyecto

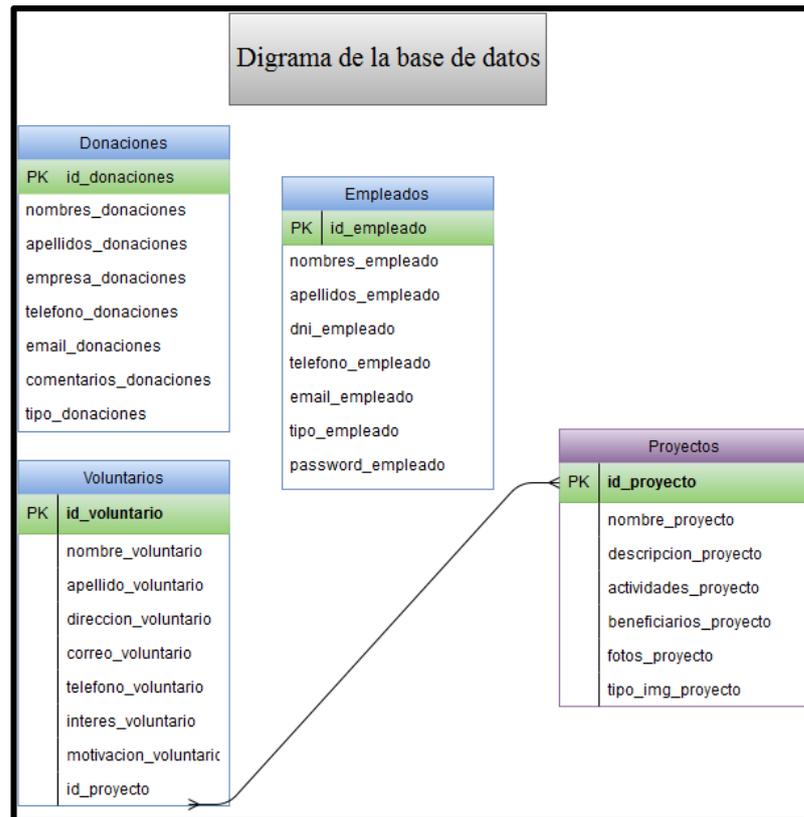


Figura No. 13. Diseño de la base de datos

Tabla No. 28.

Campos del formulario Registro voluntario

| Campo | Tipo |
|-----------------------|--------------|
| id_voluntario | int(100) KEY |
| nombre_voluntario | varchar(300) |
| apellido_voluntario | varchar(300) |
| dirección_voluntario | varchar(400) |
| profesión_voluntario | varchar(300) |
| móvil_voluntario | int(30) |
| correo_voluntario | varchar(100) |
| teléfono_voluntario | int(15) |
| interés_voluntario | varchar(300) |
| motivación_voluntario | varchar(500) |
| Id_proyecto | Int(100) |

Tabla No. 29.

Campos del formulario Registro donaciones

| CAMPO | TIPO |
|------------------------|--------------|
| id_donaciones | int(10) KEY |
| nombres_donaciones | varchar(200) |
| apellidos_donaciones | varchar(200) |
| empresa_donaciones | varchar(200) |
| teléfono_donaciones | varchar(200) |
| email_donaciones | varchar(200) |
| comentarios_donaciones | varchar(200) |
| tipo_donaciones | varchar(200) |

Tabla No. 30.

Campos del formulario Registro empleados

| CAMPO | TIPO |
|--------------------|--------------|
| id_empleado | Int(100) KEY |
| nombres_empleado | Varchar(150) |
| apellidos_empleado | Varchar(150) |
| dni_empleado | Varchar(150) |
| teléfono_empleado | Varchar(150) |
| email_empleado | Varchar(150) |
| tipo_empleado | Varchar(150) |
| passwor_empleado | Varchar(150) |

Tabla No. 31.

Campos del formulario Registro proyectos

| CAMPO | TIPO |
|------------------------|--------------|
| id_proyecto | Int(100) KEY |
| nombre_proyecto | Varchar(200) |
| descripcion_proyecto | Varchar(500) |
| actividades_proyecto | Varchar(500) |
| beneficiarios_proyecto | Varchar(400) |
| fotos_proyecto | Mediumblob |
| tipo_img_proyecto | Varchar(200) |

4.4. Arquitectura del sistema

La arquitectura del sistema MVC (Modelo, Vista, Controlador) maneja tres archivos o scripts que interactúan entre sí para el correcto funcionamiento de las capas, a continuación en el siguiente grafico vamos a representar la arquitectura del sistema.

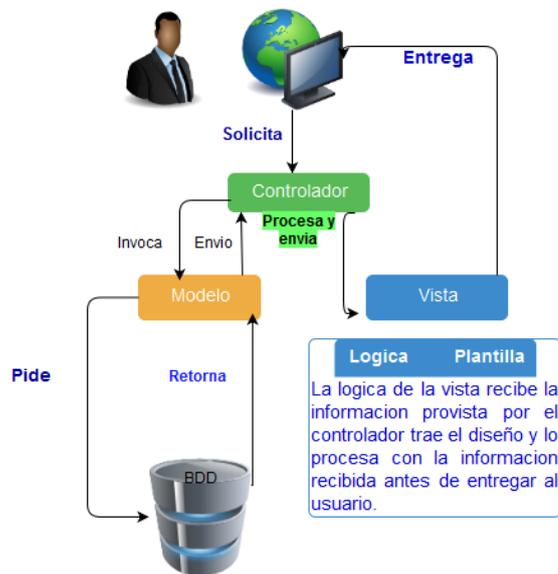


Figura No. 14. Arquitectura del sistema

4.4.1. Modelo (Model)

En esta capa el framework actúa directamente en la base de datos, haciendo las sentencias SQL más dinámicas, ya que cada sentencia de consulta SQL se encuentran definidas en clases propias de Codeigniter haciendo que tareas de codificación como consultar, insertar, recuperar, y modificar información sean más dinámicas y solo se necesite pocas líneas de código para su desempeño a continuación explicaremos la estructura y codificación de esta capa.

4.4.2. Vista (View)

Esta es la capa donde interactúa el usuario, es la parte visual del sitio web aquí tenemos los archivos de PHP que se ejecutan conjuntamente haciendo llamados a los scripts de HTML, JavaScript, CSS, etc. Es decir todo el diseño básicamente lo encontramos en esta capa, a continuación en la siguiente tabla vamos a explicar el funcionamiento de las vistas.

4.4.3. Controlador (controller)

El **controlador** hace que interactúe la capa de la vista conjuntamente con la capa del **modelo**, haciendo las peticiones de envío y recepción de datos mediante el método post o haciendo el llamado a varias funciones incluidas en las clases del controlador, generando o devolviendo la lógica del negocio por medio de la **vista**, a continuación se describen en la siguiente tabla los scripts que intervienen en esta etapa de codificación.

Tabla No. 32.

Estructura de la programación.



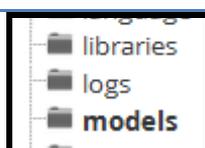
Este esquema corresponde a la estructura de la programación en capas, aquí encontramos el orden que Codeigniter maneja para trabajar en su entorno, cada script de la programación del sitio se encuentra organizado de tal manera que los cambios entre desarrolladores sean más flexibles en su búsqueda y edición. En nuestro caso nos ayuda a manejar dinámicamente la codificación, siendo más capaces de realizar cambios en los módulos desarrollados.

Nos vamos a centrar en los siguientes directorios:

- Views
- Controller
- Models

Tabla No. 33.

Codificación Modelo



En el directorio de la aplicación se encuentra la carpeta models aquí se encuentran los archivos de conexión a la base de datos.

CONTINÚA →

| | Name |
|--|------------------------------|
| | empleados_model.php |
| | index.html |
| | reporte_donacion_model.php |
| | reporte_voluntario_model.php |
| | usuario_model.php |
| | voluntarios.php |

Ubicación de todos los modelos de datos de la aplicación siguiendo una estructura y estandarización en todos los archivos, siendo más ordenada la manera de programar y facilitando la búsqueda de scripts según los mensajes de errores que se muestran en el navegador esto facilita actualizar la programación de los scripts en caso de ser necesario

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Reporte_donacion_model extends CI_Model {

    function guardar($data){
        $this->db->insert("donaciones",$data);

        if ($this->db->affected_rows() > 0) {
            return true;
        }
        else{
            return false;
        }
    }

    function mostrar($valor){
        $this->db->like("nombres_donacion",$valor);
        $consulta = $this->db->get("donaciones");
        return $consulta->result();
    }

    function actualizar($id,$data){
        $this->db->where('id_donacion', $id);
        $this->db->update('donaciones', $data);
        if ($this->db->affected_rows() > 0) {
    
```

Mediante clases y funciones definidas por Codeigniter podemos observar varias sentencias SQL, ya mencionadas anteriormente, que gestionan la información. Ejemplo: inserción de datos.
 \$this->db->insert("donaciones",\$data);

Tabla No. 34.
Vista (view)

Esta capa está organizada de manera que cada carpeta contenga los scripts del panel de administración, la parte del front-end, menús, mensajes de error y la estructura del diseño web que visualiza el usuario conjuntamente con varios menús que son nombrados en estas carpetas.

| | Name |
|--|------------------------|
| | donaciones.php |
| | donaciones_fisicas.php |
| | fotos.php |
| | hola.php |
| | index123.php |
| | login.php |

Estos Scripts manejan la parte visual del sitio web, en cada uno de ellos tenemos embebidos varios lenguajes de programación como HTML, Ajax, CSS, JavaScript, Json, que son la parte visual para el usuario, y nos brindan la navegabilidad adecuada para realizar nuestros procesos.

```

<div class="seven row">
  <div class="form-group">
    <h1>
      <span class="label label-primary">FORMULARIO PARA DONACIONES</span>
    </h1>
  </div>

  <form data-toggle="validator" role="form" name="donaciones_fisicas" method="post" >

    <div class="form-group">
      <label for="inputName" class="control-label">Nombres:</label>
      <input type="text" name="nombres_donacion" class="form-control" placeholder="Nombres" required>
    </div>

    <div class="form-group">
      <label for="inputName" class="control-label">Apellidos:</label>
      <input type="text" name="apellidos_donacion" class="form-control" placeholder="Apellidos" required>
    </div>

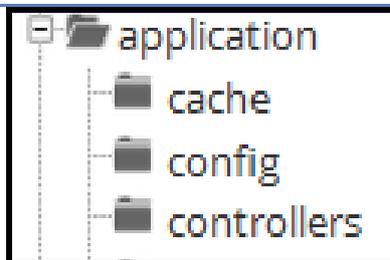
    <div class="form-group">
      <label for="inputName" class="control-label">Empresa:</label>
      <input type="text" name="empresa_donacion" class="form-control" placeholder="Empresa" required>
    </div>

    <div class="form-group">
      <label for="inputName" class="control-label">Telefono:</label>
      <input type="text" name="telefono_donacion" class="form-control" placeholder="Telefono" required>
    </div>
  </form>

```

Ejemplo de codificación de la vista, haciendo el envío del formulario de donaciones mediante el método "post" con lenguaje HTML y JavaScript embebido para la validación de campos.

Tabla No. 35.
Controlador (controller)



Esta capa está organizada de manera que cada archivo de la lógica del negocio se encuentra dentro de la carpeta controllers.

| Name |
|------------------------|
| angeles.php |
| contactanos.php |
| donaciones.php |
| donaciones_fisicas.php |
| empleados.php |
| home.php |
| index123.php |
| login.php |
| nosotros.php |
| pago.php |
| proyecto.php |

Aquí tenemos varios archivos que controlan el envío y recepción de información así como las transacciones operacionales, aritméticas, etc.

```

1 <?php
2
3 class Donaciones_fisicas extends CI_Controller
4
5 [
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

```

```

public function index()
{
    $this->load->view("guest/header");

    if (!empty($_POST))
    {
        $nombres_donacion = $_POST['nombres_donacion'];
        $apellidos_donacion = $_POST['apellidos_donacion'];
        $empresa_donacion = $_POST['empresa_donacion'];
        $telefono_donacion = $_POST['telefono_donacion'];
        $correo_donacion = $_POST['correo_donacion'];
        $comentario_donacion = $_POST['comentario_donacion'];
        $tipo_donacion = $_POST['tipo_donacion'];
    }
}

```

La codificación del Controller mediante el método post para la generación de formularios, donde las donaciones serán enviadas haciendo uso de la función load (cargar).

4.5. Diseño de las interfaces

Este diseño permite que, los requisitos levantados del sistema, se plasmen en una vista preliminar para determinar cómo está conformado el sistema, tener un punto de partida en la realización de un buen diseño que permita a los usuarios finales navegar de forma dinámica y amigable.

4.5.1. Diseño del sitio web

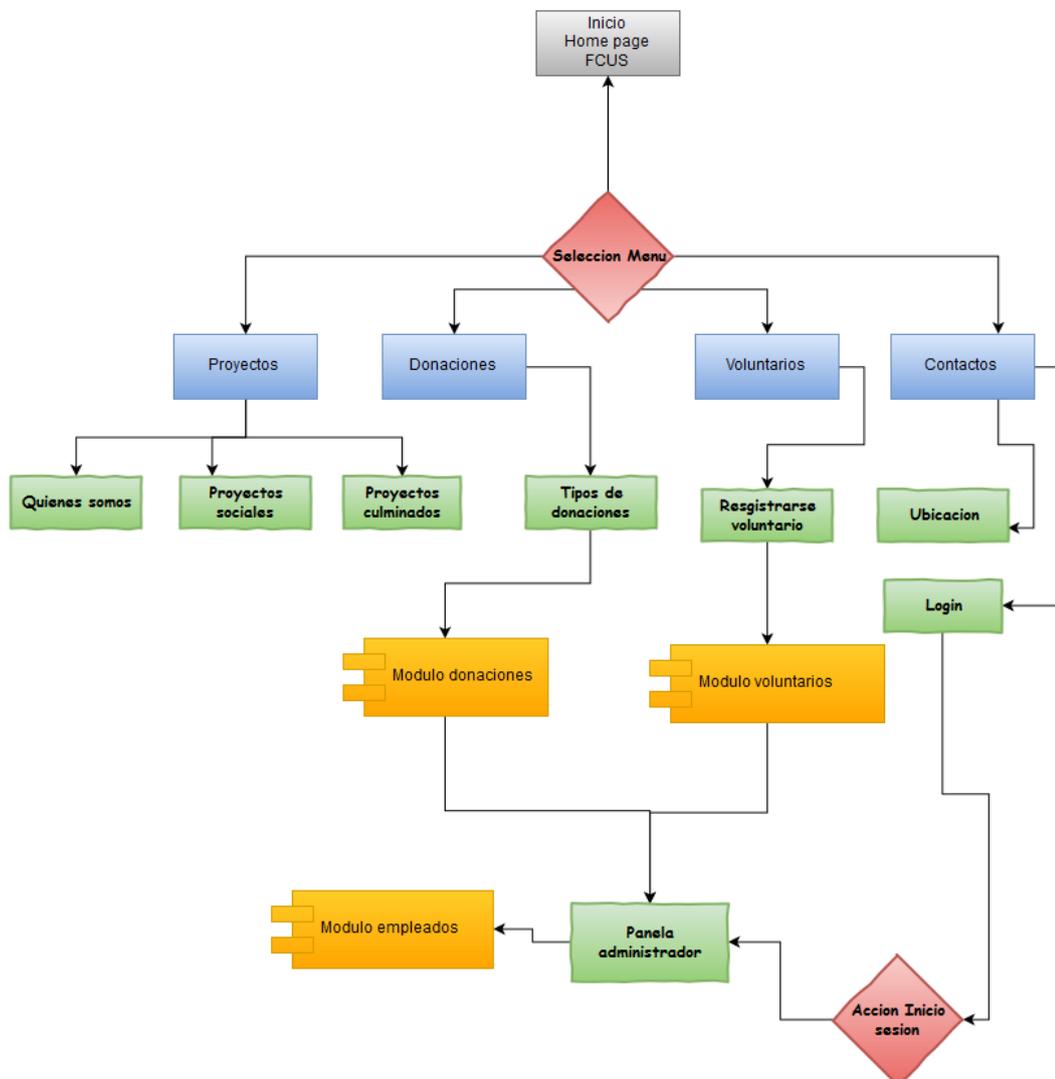


Figura No. 15. Diagrama de interfaz común



Figura No. 16. Interfaz de usuario inicial

En la interfaz inicial, el sitio web consta de 6 componentes principales.

- Área del logo de la Fundación.
- Barra de menú
- Informativo
- Accesos directos a la funciones principales
- Banner de proyectos
- Sección de imágenes auspiciantes.
- Pie del sitio

Pestaña ¿Quiénes somos?

Aquí se describe la Misión, Visión, e Historia de la fundación.



Figura No. 17. Interfaz historia de la fundación

Pestaña Proyectos sociales

Aquí encontramos los proyectos que actualmente la Fundación está llevando a cabo.



Figura No. 18. Interfaz proyectos sociales

Pestaña Proyectos culminados

Aquí encontramos los proyectos culminados de la Fundación



Figura No. 19. Interfaz proyectos culminados

Pestaña Voluntario

Aquí encontramos el formulario de registro para los usuarios del sitio web de la Fundación que deseen registrarse como voluntarios.



Figura No. 20. Interfaz registro voluntario

Pestaña Donaciones

Aquí encontramos el registro de donaciones para la Fundación



Figura No. 21. Interfaz registro donación

Pestaña Contactos

Aquí encontramos el mapa de Google que nos muestra la ubicación de la Fundación.



Figura No. 22. Interfaz ubicación

Pestaña Login

Aquí encontramos un link para la ventana de Iniciar sesión para el administrador del sitio web.

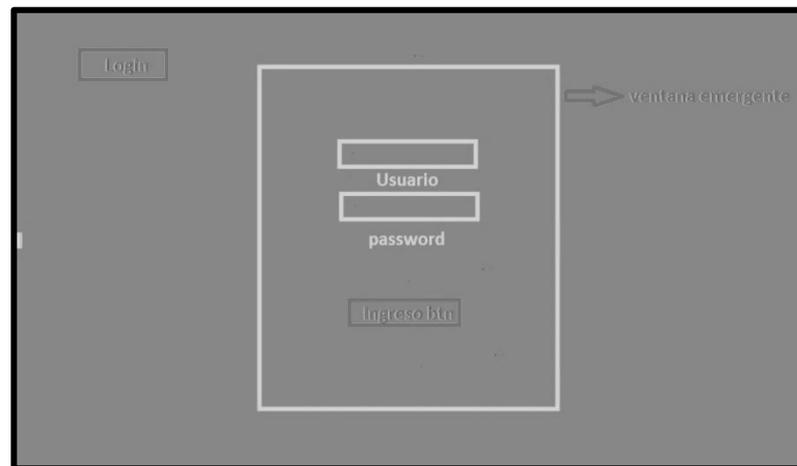


Figura No. 23. Interfaz de inicio de sesión

Panel de control del Administrador del sitio

Aquí encontramos a las gestiones del sistema que permiten al administrador del sitio, realizar la manipulación de información.

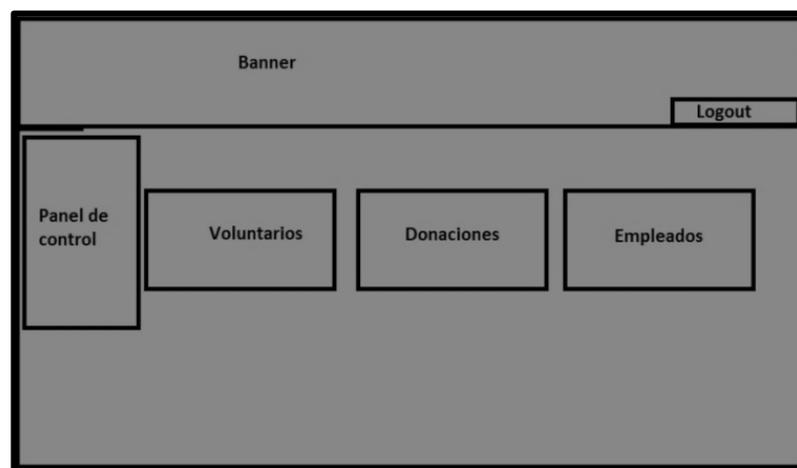


Figura No. 24. Panel administrador

4.6. Implementación

Procedimiento de instalación para la implementación de nuestro sistema, subimos el sitio web al internet, una vez ya cumplido con todos los requisitos del desarrollo del sitio web.

Es necesario leer el manual de configuración del sistema que se encuentra en el anexo 3, para la correcta implementación tomamos en cuenta que esta arquitectura cliente servidor presenta 3 capas, por lo que la configuración de la base de datos es de vital importancia en este proceso, se necesitan las siguientes características para levantar la base de datos y poner en producción al sistema web.

- Servidor web apache
- Script SQL de la base de datos FCUS
- Dominio web
- phpMyAdmin

Ingresamos a la página de phpMyAdmin en el panel de control del servidor web.



Figura No. 25. Panel de control servidor Web

Una vez que nos ubicamos en el panel de control de la base de datos, vamos a la pestaña Importando al servidor, que permite la importación del archivo SQL, es decir el script de la base de datos.

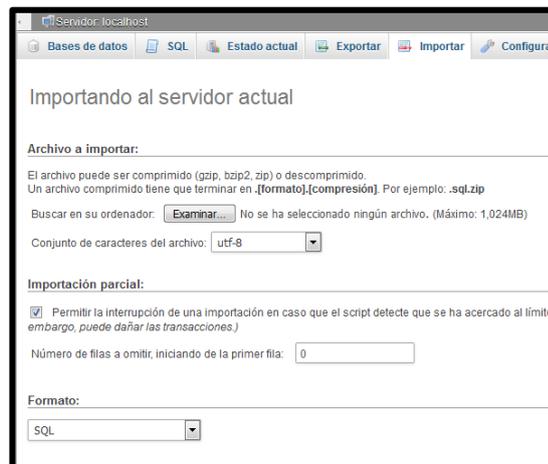


Figura No. 26. Panel de importación de la base de datos

Con la estructura de datos en el servidor web, podremos proceder a seguir los pasos del manual de configuración del sistema que se encuentra en el anexo 3.



Figura No. 27. Estructura de la base de datos MYSQL

Una vez realizados estos pasos, el sitio está listo para la navegación de la interfaz.

4.7. Codificación Framework Codeigniter (MVC)

En el capítulo 3 seleccionamos a Codeigniter como tecnología de desarrollo MVC (Modelo, vista, controlador), este framework PHP, permitirá la creación de módulos dinámicos para la reutilización del código en las clases que posee por defecto, las más importantes vamos a mencionarlas a continuación describiendo cada una de las capas codificadas para nuestro sitio web.

Para el procesado de datos y presentación del sitio, utilizamos varios lenguajes de programación como es CSS, JavaScript, HTML. Todos estos archivos se encuentran en el directorio del servidor Apache.

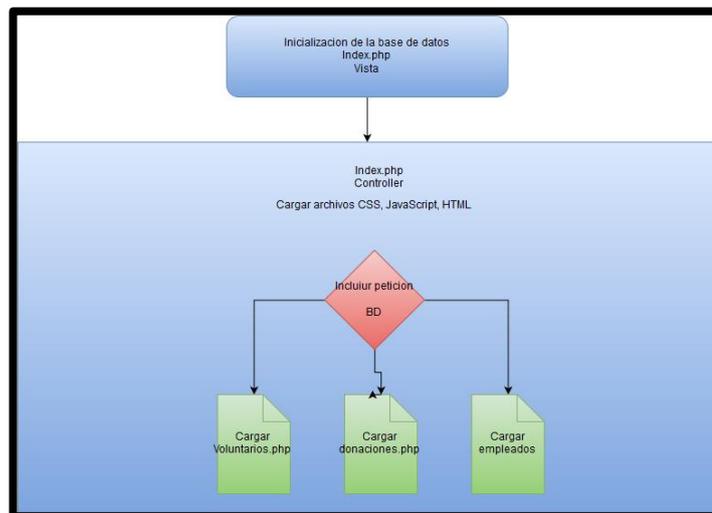


Figura No. 28. Interfaz de datos Fuente: Obando 2016

4.8. Pruebas del sistema

Estas pruebas del sistema permiten verificar falencias y errores relacionados con la navegación, implementación y usabilidad, se tomarán en cuenta pruebas de interfaz así como también pruebas en el envío y recepción de datos para mejorar la codificación si es necesario. Esto consiste en verificar que los requisitos planteados en las definiciones de funcionalidad sean correctos, durante la implementación se han ido realizando las pruebas correspondientes, a continuación revisamos dichas pruebas.

- Pruebas de interfaz y contenido
- Pruebas funcionales y operacionales

4.8.1. Pruebas de interfaz y contenido.

El objetivo de este plan de pruebas está orientado a la verificación de contenidos, mediante distintos browsers como FireFox, Crome y Opera.

| Pruebas de interfaz | |
|--|-------------------|
| No 1 | Acceso al sistema |
| Condiciones: | |
| <ul style="list-style-type: none"> • El usuario debe tener un navegador web • Debe existir conexión a internet | |
| Entrada: | |
| <ul style="list-style-type: none"> • El usuario debe acceder al link: www.fcus-ec.org | |
| Resultados: | |
| <ul style="list-style-type: none"> • Ingreso exitoso, se despliega el Home Page | |
| Evaluación de la prueba: | |
| <ul style="list-style-type: none"> • Satisfactoria | |

| Pruebas de interfaz | |
|--|----------------------------|
| No 2 | Verificación del contenido |
| Condiciones: | |
| <ul style="list-style-type: none"> • Conexión a internet • Navegador web | |
| Entrada: | |
| <ul style="list-style-type: none"> • El usuario debe acceder al link: www.fcus-ec.org | |
| Resultados: El contenido web que se describió en los requisitos de interfaz son los definidos por el cliente. | |
| Evaluación de la prueba: Satisfactoria. | |

| Pruebas de interfaz | |
|---|--|
| No 3 | Verificación de links vacíos o en construcción |
| Condiciones: | |
| <ul style="list-style-type: none"> • Conexión a internet • Navegador web | |
| Entrada: | |
| <ul style="list-style-type: none"> • El usuario debe acceder al link: www.fcus-ec.org • Validación de pestañas • Validación de diseño de formularios • Validación de acceso al panel | |
| Resultados: No existe ningún link vacío | |
| Evaluación de la prueba: Satisfactoria | |

| Pruebas de interfaz | |
|---|---------------------|
| No 4 | Validación HTML CSS |
| Condiciones: | |
| <ul style="list-style-type: none"> • Conexión a internet • Navegador web | |
| Entrada: | |
| <ul style="list-style-type: none"> • El usuario debe acceder al link: www.fcus-ec.org • Verificar tiempos de respuesta de los archivos HTML y CSS se carguen | |

CONTINÚA →

correctamente.

- Realizar un seguimiento de la navegación jerárquica definida sea la correcta.

Resultados: **Sigue los diseños definidos por los requisitos**

Evaluación de la prueba: **Satisfactoria**

4.8.2. Pruebas funcionales y operacionales

Estas pruebas están orientadas a la comprobación de funcionalidades de la aplicación, pruebas de formularios que envían y reciben información, validación de campos obligatorios y las consultas realizadas que deben ser completas y correctas.

Pruebas funcionales y operacionales

No 5 Campos obligatorios envío y recepción

Condiciones:

- Conexión a internet
- Navegador web
- Completar los formularios con los datos solicitados

Entrada: **El usuario debe acceder a los links:**

- www.fcus-ec.org/Voluntario
- www.fcus-ec.org/Donaciones_fisicas
- www.fcus-ec.org/Login

Aquí se encuentran los formularios de registro, el usuario debe llenar los campos requeridos los cuales cambian a color rojo al considerarse campos obligatorios.

Resultados: **En los tres casos se refleja las operaciones exitosas y se puede observar las validaciones respectivas.**

Evaluación de la prueba: **Satisfactoria**

Pruebas funcionales y operacionales

No 6 Validación formularios administrador

Condiciones:

- Ingresar como administrador
- Conexión a internet
- Navegador web
- Llenar los formularios

Entrada:

- El usuario debe acceder al link: www.fcus-ec.org/Login
- Ingreso de información en los formularios.
- Verificación de datos de perfil administrador.
- Verificación de los botones de ingreso, eliminación, y edición de los datos que tienen registrado las diferentes gestiones como Voluntarios, Donaciones, Empleados.

Resultados: **Los módulos de las gestiones como Voluntarios, Donaciones, Empleados fueron verificadas y comprobadas correctamente, todas las funcionalidades correspondientes al administrador del sitio están verificadas.**

Evaluación de la prueba: **Satisfactoria**

Pruebas funcionales y operacionales**No 4 Elementos de interfaz**

Condiciones:

- **Conexión a internet**
- **Navegador web**

Entrada:

- **El usuario debe acceder al link: www.fcus-ec.org/index123**
- **Verificar si los elementos de la interfaz tienen una apariencia uniforme mediante el uso de plantillas dinámicas.**

Resultados: **Resultados correctos se encuentran con el mismo formato de registro, consulta y edición en la definición del panel de control.**

Evaluación de la prueba: **Satisfactoria**

Pruebas funcionales y operacionales**No 4 Elementos de interfaz**

Condiciones:

- **Conexión a internet**
- **Navegador web**

Entrada:

- **El usuario debe acceder al link: www.fcus-ec.org/index123**
- **Verificar si los elementos de la interfaz tienen una apariencia uniforme mediante el uso de plantillas dinámicas.**

Resultados: **Resultados correctos se encuentran con el mismo formato de registro, consulta y edición en la definición del panel de control.**

Evaluación de la prueba: **Satisfactoria**

CAPÍTULO V

5. VALIDACIÓN DE LA PROPUESTA

5.1. Entrega del sistema

5.1.1. Descripción del proyecto

Este proyecto tiene como objetivo fomentar la participación ciudadana, por medio del sitio web vinculándolos directamente en los diferentes proyectos de ayuda social que brinda la Fundación “Construyamos un sueño”, ubicada en la ciudad de Latacunga, el sitio web www.fcus-ec.org, gestionará la aportación económica mediante PAYPAL. Una vez concluido el desarrollo del mismo, permitirá que el personal de FCUS, lleve el control de nuevos voluntarios y donaciones, esto hace que estos procesos sean efectivos ya que se los manejaba de forma manual.

5.1.2. Alcance del proyecto

El alcance del proyecto se centra en cuatro puntos principales:

- Diseño del sitio web.
- Gestión de voluntarios.
- Gestión de empleados.
- Gestión de donaciones.

5.1.3. Supuestos

- Se cuenta con disponibilidad de fondos para la ejecución del proyecto
- Se cuenta con personal técnico y administrativo para la implementación del proyecto

- Existe la apertura del personal involucrado en el proyecto para generar los diferentes requisitos funcionales del proceso
- Se cuenta con la infraestructura de Equipos y Servidores disponibles para la implementación del proyecto.
- Se cuenta con el personal técnico y usuarios finales que serán capacitados para el manejo del Sistema.

5.1.4. Restricciones

- El proyecto deberá ejecutarse en un plazo de 120 días
- El presupuesto del proyecto es de 120 USD

5.1.5. Requerimientos del proyecto

A continuación, se detallan los requerimientos del proyecto:

5.1.6. Requerimientos Generales

- Implementar un sitio web que gestione los registros de voluntarios y donaciones físicas y económicas
- Crear un módulo para donaciones mediante PayPal.
- Crear un panel de control de administrador

5.1.7. Requerimientos Específicos

- El sitio tendrá dos ambientes el de usuario normal y el de administrador.
- Se crearán un formulario de búsquedas en el panel de administrador con los campos relacionado con los índices definidos con el fin de que el usuario ubique fácilmente la información registrada.
- Se podrán registrar los usuarios, donaciones, empleados por medio del sitio.

5.1.8. Requerimientos no funcionales

- Acceso al sistema mediante cualquier navegador de Internet.
- De fácil uso para las personas que quieren registrarse y realizar donaciones.
- Sistema atractivo, usable, rápido y confiable.
- El rendimiento de la base de datos debe ser óptimo en el tiempo de transacciones propias del sistema.
- La información manejada por el sistema está protegida de acceso no autorizado y divulgación.

Estos requisitos se cumplieron con todas las funcionalidades necesarias, estando presente la representante legal de la fundación en la ciudad de Latacunga se hace la entrega del sistema web cuyo objetivo fundamental es mejorar los procesos de registro de voluntarios y donaciones de la fundación por lo que se cuenta con el documento de respaldo de la entrega del sistema

5.2. Aceptación del sistema.

Se hace la entrega del sitio web www.fcus-ec.org a la Fundación “Construyamos un sueño” con las características y funcionalidades definidas en el proyecto, por lo que la fundación respalda el desarrollo del sitio web con una carta de aceptación. El documento al que hace referencia este titular se encuentra en la sección del anexo A.

5.3. Manual de usuario.

El documento al que hace referencia este titular se encuentra en la sección del anexo B.

5.4. Manual de configuración del sistema.

El documento al que hace referencia este titular se encuentra en la sección del anexo B.

5.5. Conclusión del capítulo.

Podemos concluir en este capítulo, que nos permitió implementar positivamente la mejor opción de desarrollo ágil y su mejor plataforma en Frameworks PHP los cuales son Scrum y Codeigniter respectivamente, ya que se ajustó los requisitos del sistema y tiempos para el desarrollo del mismo, frente al resto de tecnologías, si se trata de proyectos pequeños estas nos facilitan el desarrollo. La aceptación del sitio por parte de la Fundación “Construyamos un Sueño” fue inmediata y positiva, ya que logró ajustarse a cronogramas, plazos y recursos disponibles lo que permitió la satisfacción del cliente en la entrega final.

CAPÍTULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- A partir del desarrollo de la propuesta planteada se pudo establecer que, la definición de un marco teórico que reúna la información para el desarrollo del proyecto, es de vital importancia para lograr una adecuada selección de la mejor tecnología encaminando nuestro proyecto de una manera clara y concisa.
- Se pudo identificar deficientes flujos de trabajo para el registro de donaciones en la fundación, por lo que el desarrollo de la propuesta brindará una mejora significativa en dicho proceso.
- Se concluye, a partir del desarrollo del Sitio web, que la información manejada dentro del mismo, responde de manera directa a la documentación entregada por la fundación, recolectando los datos mediante la creación de formularios, exitosamente.
- Mediante el desarrollo del presente proyecto se concluye que, la implementación de la propuesta generó en la fundación, una mejor comunicación con los clientes, por lo que el incremento en las donaciones económicas fue notable, ya que se realizan por medio de la página web.
- Podemos concluir que la investigación y comparación de metodologías de desarrollo ágil, así como también las tecnologías PHP que fomentan las correctas prácticas de diseño, modelo, vista, controlador, fueron de vital importancia para la implementación del sitio.
- La utilización de la metodología ágil SCRUM, nos brindó el soporte necesario para poder cumplir con los tiempos de entrega del sistema web.

- La codificación mediante el framework Codeigniter, nos ayudó a la correcta programación aplicando estándares y brindando la seguridad necesaria para el envío de datos.

6.2. Recomendaciones

- Se recomienda que a las personas que se vinculen con el proyecto o que estén involucradas en la nueva versión de sitio web, deben leer este documento para conocer de forma clara el proceso de desarrollo del mismo.
- Se recomienda el estudio detallado del framework “Codeigniter”, ya que permitirá establecer una correcta configuración de la página, siguiendo los lineamientos y recomendaciones en cuanto a la seguridad de la información del framework.
- Se recomienda que el código HTML del botón “PayPal”, vinculado al sitio web al que hace referencia este documento no debe ser modificada bajo ninguna circunstancia ya que cuenta con firmas electrónicas propias de PayPal y perdería su funcionamiento.
- Las herramientas y metodología utilizadas en este proyecto brindaron mayor aceptación al cambio constante de requisitos y a buenas prácticas en la programación en capas, por lo que son adecuadas para utilizar en proyectos pequeños.
- En caso de que se requiera expandir el sistema, recomendamos seguir la estructura MVC, ya que tendrá un mejor mantenimiento y orden al implementarlo.
- Se puede añadir otros módulos a futuro, podrían considerarse gestionar la parte administrativa del personal, además de la creación del módulo de contabilidad.

REFERENCIAS BIBLIOGRÁFICAS

- Acuña, K. B. (06 de 06 de 2010). *enumed.net*. Recuperado el 23 de 06 de 2016, de enumed: <http://www.eumed.net/libros-gratis/2009c/584/Metodologias%20tradicionales%20y%20metodologias%20agiles.htm>
- Álvarez Santos, A., & Gómez Pinto, A. J. (2015). *Sistema de registro único de trámites y servicios municipales*. Chiapas, México DF: Instituto Tecnológico de Tuxtla Gutiérrez. Recuperado el 23 de 12 de 2015
- Álvarez, M. Á. (02 de Enero de 2014). *Qué es MVC*. Recuperado el 02 de 08 de 2015 Obtenido de Desarrollo Web: <http://www.desarrolloweb.com/articulos/que-es-mvc.html>
- Beck, K. (1999). *Extreme Programming Explained. Embrace Change*. Pearson Education. Recuperado el 11 de 06 de 2016
- Blogger, J. (10 de Marzo de 2010). *Modelado del análisis para aplicaciones web*. Recuperado el Marzo de 2016, de <http://elchrboy.blogspot.com/2010/03/modelado-del-analisis-para-aplicaciones.html>
- Bonilla Orozco, J. P. (16 de Abril de 2014). *Trabajo de Proyecto de Corte*. Recuperado el 08 de Febrero de 2016, de Blogspot: <http://bonilla-jessica.blogspot.com/>
- Cárdenas, H. (2015). *Postgre SQL*. Obtenido de http://postgresql2013.blogspot.com/p/introduccion_11.html Recuperado el 28 de 12 de 2015
- Cases, E. F. (21 de Septiembre de 2014). *ibrugor*. Recuperado el 2 de Febrero de 2016, de <http://www.ibrugor.com/blog/que-es-php-para-que-sirve/>
- CCM. (2 de Octubre de 2015). *Intranet y extranet*. Recuperado el 2 de Febrero de 2016, de <http://es.ccm.net/contents/213-intranet-y-extranet>
- Cruz Catacora, E. A. (25 de Septiembre de 2012). *Ingeniería Web*. Obtenido de Slide Share: <http://es.slideshare.net/edwardcruz9638/ingenieria-web-14457826> Recuperado el 9 de 07 de 2016
- DE. (2015). *Diferencia entre Sitio Web y Página Web*. Obtenido de Diferencias entre: <http://www.diferencia-entre.com/diferencia-entre-sitio-web-y-pagina-web/> Recuperado el 14 de 05 de 2015

- Dirección de Comunicación Universitaria Departamento de Prensa. (11 de Junio de 2012). *Programación extrema: "Metodología para desarrollo ágil de aplicaciones"*. Obtenido de Universo, el periódico de los universitarios: http://www.uv.mx/universo/486/infgral/infgral_15.html Recuperado el 23 de 06 de 2014
- Duarte, E. (19 de Octubre de 2012). *¿Qué Es JavaScript ? Ventajas Y Desventajas*. Obtenido de Capacity: <http://blog.capacityacademy.com/2012/10/19/que-es-javascript-ventajas-y-desventajas/> Recuperado el 02 de Agosto de 2015
- González, J. (2 de Septiembre de 2013). *Host dime blog*. Recuperado el 2 de Febrero de 2016, de <http://blog.hostdime.com.co/6-frameworks-php-para-el-desarrollo-agil-de-aplicaciones-web/> Recuperado el 23 de Diciembre de 2015
- Guerrero, J. (28 de Junio de 2008). *Metodologías Ágiles de desarrollo de software (XP) Fases*. Obtenido de http://boards5.melodysoft.com/UBV_INGS/metodologias-agiles-de-desarrollo-43.html Recuperado el 23 de 09 de 2016
- Hassan, Y., Francisco, t., Fernández, M., & Iazza, G. (1 de Febrero de 2004). *Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información*. Recuperado el 10 de Marzo de 2016, de https://www.upf.edu/hipertextnet/numero-2/disenio_web.html
- Ing. Software (Equipo 02). (08 de Julio de 2012). *Metodología Extreme Programming XP*. Obtenido de <http://ingsoftware072301.obolog.es/metodologia-xp-2012877> Recuperado el 13 de 04 de 2016
- Jordisan.net. (29 de Septiembre de 2006). *¿Qué es un 'framework'?* Recuperado el 2 de Febrero de 2016, de <http://jordisan.net/blog/2006/que-es-un-framework/> Recuperado el 12 de 05 de 2016
- Kinderman , H., Silva, C., & Velásquez, V. (2015). *PostGreSQL*. Recuperado el Enero de 2016, de <http://postgresql-adsi.blogspot.com/2011/11/ampliamente-popular-ideal-para.html> Recuperado el 17 de 06 de 2016

- Kinderman, H., Silva, C., & Velásquez, V. (26 de Noviembre de 2011). *PostGreSQL*. Recuperado el 2 de Febrero de 2016, de <http://postgresql-adsi.blogspot.com/2011/11/ampliamente-popular-ideal-para.html>
- La Info. (2 de Agosto de 2013). *superinformacionweb*. Recuperado el 2 de Febrero de 2016, de <http://superinformacionweb.blogspot.com/2014/11/ventajas-y-desventajas-de-mysql.html>
- Lab. (2 de Febrero de 2013). *Patrón de arquitectura Modelo Vista Controlador (MVC)*. Recuperado el 2 de Febrero de 2016, de EjemplosTIW: <http://www.lab.inf.uc3m.es/~a0080802/RAI/mvc.html>
- Lamarca Lapuente, M. J. (08 de Diciembre de 2013). *Diseño de una página web*. Obtenido de Hipertexto: http://www.hipertexto.info/documentos/diseg_web.htm Recuperado el 23 de febrero de 2015
- Lenguaje de programación y lenguajes de etiqueta*. (20 de Enero de 2014). Obtenido de Blog de WordPress.com.: <https://eduherramientasweb20.wordpress.com/tag/lenguaje-de-programacion-y-lenguajes-de-etiqueta/> Recuperado el 14 de 03 de 2016
- Letelier, P., & Penadés, M. C. (Abril/Junio de 2006). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Recuperado el Febrero de 2016, de Técnica Administrativa, Buenos Aires: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
- Libros web. (2 de Septiembre de 2007). *HTML*. Recuperado el 2 de Febrero de 2016, de <http://librosweb.es/libro/xhtml/>
- Libros web. (2 de Febrero de 2014). *La Arquitectura MVC*. Recuperado el 2 de Febrero de 2016, de http://librosweb.es/libro/jobeeet_1_4/capitulo_4/la_arquitectura_mvc.html
- Maigua Soliz, A. M. (15 de Junio de 2009). *Ingeniería Web*. Obtenido de Mi Blog Ana: <http://anamaigua.blogspot.com/2009/06/ingenieria-web.html> Recuperado el 23 de Septiembre de 2015
- Mirsha. (29 de Agosto de 2013). *Ingeniería Web*. Obtenido de Slide Share: <http://es.slideshare.net/Mirsha01/ingenieria-web-25718669>

- MySQL. (2015). *Rendimiento MYSQL*. Obtenido de Ventajas y Desventajas: <https://mysqldaniel.wordpress.com/ventajas-y-desventajas/>
Recuperado el 03 de Marzo de 2016
- Paginas web. (02 de Febrero de 2014). *Diferencias*. Recuperado el 02 de Febrero de 2016, de <http://www.diferencia-entre.com/diferencia-entre-sitio-web-y-pagina-web/>
- Pérez Valdés, D. (02 de Noviembre de 2007). *Los diferentes lenguajes de programación para la web*. Obtenido de Maestros del web: <http://www.maestrosdelweb.com/los-diferentes-lenguajes-de-programacion-para-la-web/> Recuperado el 23 de Abril de 2015
- Pérez, J. L. (16 de Septiembre de 2015). *Principales lenguajes de programación web, ventajas y desventajas*. Obtenido de Registro y registradores de dominios: <http://www.registrodominiosinternet.es/2013/08/lenguajes-programacion-web-ventajas.html> Recuperado el 23 de Junio de 2015
- PostgreSQL. (5 de Febrero de 2014). *Características, limitaciones y ventajas*. Recuperado el 2 de Febrero de 2016, de PostgreSQL: <http://postgresql-dbms.blogspot.com/p/limitaciones-puntos-de-recuperacion.html> Recuperado el 23 de Julio de 2015
- Pressman, R. S. (2011). *Libro de Ingeniería del Software (Un Enfoque Práctico)*. Recuperado el 15 Febrero de 2016, de <http://player.slideplayer.es/17/3427344/data/images/img3.jpg>
- Ramos, M. R. (15 de Junio de 2015). *el web master*. Recuperado el 2 de Febrero de 2016, de <http://www.elwebmaster.com/general/los-10-mejores-frameworks-de-php-para-el-2015>
- Rangel, K. (04 de Mayo de 2009). *Ingeniería Web*. Recuperado el 03 de Octubre de 2016, de Slide Share: <http://es.slideshare.net/karenx28/ingenieria-web-1382399>
- Scrum Alliance. (2016). *Faced with a complex project? Use Scrum to improve teamwork, communications, and speed. See how*. Obtenido de http://www.scrumalliance.org/pages/what_is_scrum

- Scrum manager. (13 de Agosto de 2013). *Sprint*. Recuperado el 2 de Febrero de 2016, de Ciclo de tiempo en el que se desarrolla cada incremento iterativo del producto: <http://www.scrummanager.net/bok/index.php?title=Sprint>
- Scrum manager. (2 de Febrero de 2015). *Scrum*. Recuperado el 2 de Febrero de 2016, de http://www.scrummanager.net/bok/images/thumb/0/0e/Historia_usuario_ejemplo_1.jpg/800px-Historia_usuario_ejemplo_1.jpg
- Scrum manager. (2 de Febrero de 2015). *Scrum*. Recuperado el 2 de Febrero de 2016, de Roles: http://www.scrummanager.net/bok/index.php?title=File:Comprometidos_implicados.png
- Scrum manager.net. (18 de Mayo de 2014). *Artefactos Sprint*. Recuperado el 02 de Marzo de 2016, de <http://www.scrummanager.net/bok/index.php?title=Artefactos>
- Slide Share. (2 de Febrero de 2014). *Etapas de la ingeniería web*. Recuperado el 2 de Febrero de 2016, de <http://image.slidesharecdn.com/ingenieriaweb-120518174003-phpapp01/95/ingenieria-web-6-1024.jpg?cb=1337362871>
- Slideplayer. (2 de Febrero de 2015). *Fases*. Recuperado el 2 de Febrero de 2016, de XP Aplicado: http://images.slideplayer.es/8/2273638/slides/slide_21.jpg
- Smanager. (13 de Agosto de 2013). *Ejemplo pila producto*. Obtenido de http://www.scrummanager.net/bok/index.php?title=File:Ejemplo_pila_producto.png Recuperado el 28 de Agosto de 2016
- Smanager. (13 de Agosto de 2013). *Modelo evolutivo iterativo*. Obtenido de http://www.scrummanager.net/bok/index.php?title=File:Modelo_evolutivo_iterativo.png Recuperado el 10 de Diciembre de 2015
- Softeng. (2015). *Proceso y Roles de Scrum*. Obtenido de The internet development company: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html> Recuperado el 14 de Febrero de 2015

- Scrum manager. (2 de Febrero de 2014). *Scrummanager.net*. Recuperado el 2 de Febrero de 2016, de http://www.scrummanager.net/bok/images/9/90/Requisitos_agiles.png
- Taranto Vera, G. J., & Baquerizo Palma, G. (2003). *Diseño y elaboración de un sistema de información para el manejo de historias clínicas de pacientes con cáncer en los hospitales de la ciudad de Guayaquil*. Guayaquil: Escuela Superior Politécnica del Litoral. Recuperado el 23 de Julio de 2016
- Tripod. (2015). *Fases de la Programación Extrema*. Obtenido de <http://programacionextrema.tripod.com/fases.htm>
- tuartdecors.blogspot.com. (23 de Junio de 2015). *Ventajas y desventajas además mapa mental explicando MVC de forma interactiva*. Obtenido de Framework: <http://frameworkmvc.blogspot.com/2015/06/mapa-mental-explicando-mvc.html> Recuperado el 23 de Diciembre de 2015
- Valdivieso, D. (14 de Junio de 2007). *Proceso De La Ingeniería Web*. Obtenido de Ingeniería Web: <https://sistemas3.wordpress.com/2007/06/14/proceso-de-la-ingenieria-web/> Recuperado el 20 de Septiembre de 2016
- Webs Uvigo. (2015). *Server*. Recuperado el Febrero de 2016, de <http://webs.uvigo.es/redes/images/Server.jpg>

ANEXOS

Anexo A.
Aceptación del sistema

Anexo B.

**Manual de usuario y
configuración**

Anexo C.
Ficha de voluntario



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SOFTWARE**

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por el señor,
DANIEL ALEJANDRO OBANDO MOSQUERA.

En la ciudad de Latacunga, a los 03 días del mes de agosto del 2016.

Ing. Ivone Arias

DIRECTOR DEL PROYECTO

Aprobado por:

Ing. Lucas Garces

DIRECTOR DE CARRERA

Ing. Rodrigo Vaca

SECRETARIO ACADÉMICO