

Universidad de las Ciencias Informáticas

Facultad 8



# Sistema de Gestión y Control de Métricas

---

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas.



**Autores:**

Leyvis Luis Valdes López  
Andry Leal Ortiz

**Tutores:**

Ing. Yunesti Pérez La Rosa  
Ing. Irina Napal Torres

Ciudad de la Habana. Junio 2008

“Año 50 de la Revolución”

---

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Autores:

Leyvis Luis Valdes López

Andry Leal Ortiz

\_\_\_\_\_

\_\_\_\_\_

Firma del Autor

Firma del Tutor

Tutores:

Ing. Yunesti Pérez La Rosa

Ing. Irina Napal Torres

\_\_\_\_\_

\_\_\_\_\_

Firma del Autor

Firma del Tutor

---

*«Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.»*

*Kent Beck.*

## AGRADECIMIENTOS

*A nuestros padres y abuelos nadie más que ellos hacen realidad el hecho de vernos hechos ingenieros.*

*A nuestras familias por sabernos guiar en la vida por el camino correcto.*

*A nuestro Comandante en Jefe Fidel Castro Ruz,  
por haber hecho realidad el sueño de ser un profesional.*

*A nuestros amigos.*

*A nuestros profesores, indudables creadores de lo que hemos podido llegar a ser.*

*A nuestros compañeros de grupo y de proyecto.*

*A tutores, en especial a Yunesti Pérez La Rosa, por su ayuda.*

*A la UCI por ser nuestro hogar en los últimos cinco años de nuestras vidas.*

*A todos aquellos que no hemos mencionado, pero que de sobra se deben dar por aludidos por haber contribuido  
de un modo u otro en nuestra formación y a la consumación de este sueño.*

*Le agradezco:*

*A mis padres, Yolanda y Luis Oscar que siempre han estado a mi lado y que tanto me han enseñado en esta vida, a los cuales debo lo que soy.*

*A mi Hermano que me ha ayudado todo este tiempo y me ha guiado en la vida.*

*En especial a mi abuela Amparo por quererme tanto y estar siempre pendiente de mí y a mi abuelo Luis que nunca me olvidaré de él.*

*A Yole y Hilda por ser tan buenas conmigo.*

*A mis amigos de la Universidad como Humbe, Ulises, Pablo, Nordy, Henry Herminio, Yendry, Julio, Alexander, el gordo Derlys, Aldo, Andry, y a todos los que de una forma u otra han demostrado su amistad.*

*A otros como Liván, Zoilita, Yordán, Yolandita, Yadira.*

*A Yendy por acompañarme y ayudarme estos últimos meses.*

*A mis profesores de Universidad por enseñarme todo lo que sé. En fin, a todas las personas que de alguna forma han ocupado un lugar especial en mi vida.*

*Leyvis Luis*

*Les agradezco:*

*A mis abuelos Reinaldo y Caridad por ser los principales responsables de que haya cumplido este gran sueño, sin su apoyo incondicional, amor y entrega completa no hubiera podido llegar hasta donde estoy hoy, gracias por ser los mejores, todas las palabras del mundo no serían suficientes para expresarles cuanto los amo y admiro, gracias por estar ahí, justo donde y cuando más los necesito.*

*A mis padres Julio y Clara por haberme guiado siempre por el camino correcto.*

*A mi hermana Annialis que a pesar de las riñas y las discusiones la quiero con la vida.*

*A mi tía Yamileidys y mi prima Melissa, gracias por estar siempre pendientes de mí y brindarme su ayuda en todo momento.*

*A Arianna, gracias por estar a mi lado en estos meses finales y brindarme su apoyo incondicional.*

*A mis vecinos Ricardo, Mariela, Ricardín, Maritzita y Yumisleidys que todo este tiempo han sido para mí como familia.*

*A Yoilen por estar junto a mí en las buenas y las malas durante estos 5 años.*

*A todos mis amigos, Lisbel, Mario Orlando, Luis Daniel, Henry Herminio, Yendry, Ulises, Pablo, Nordy, Humberto, Leyvis, a Derlys por hacerme las noches de laboratorio tan amenas con sus pesadeces.*

*Andry*

**DEDICATORIA**

*A mi Mamá, mi Papá, mi Abuela y a mi Hermano.*

*Leyvis Luis*

*A mis Abuelos, mis padres, mi hermana, mi tía y mi prima.*

*Andry*

---

## RESUMEN

En la actualidad la calidad del software es un tema polémico y es necesario saber definir adecuadamente la eficacia de un sistema informático y cómo evaluarla mediante la utilización correcta de distintas herramientas, la calidad del software debe tenerse en cuenta en todas las etapas del desarrollo de la aplicación, con el objetivo de satisfacer las necesidades de los clientes. La posibilidad de medir es el fundamento de las disciplinas científicas y de ingeniería, ya que no hay forma de controlar lo que no se mide. Las Métricas ayudan en la estimación, el control de calidad, la evaluación de productividad y el control de proyectos. La herramienta informática propuesta permite llevar a cabo un control sistemático de cada uno de los proyectos productivos existentes en la Universidad de Ciencias Informáticas, dándole la posibilidad al personal encargado, perteneciente al Departamento Central de Calidad, de introducir métricas en el mismo, con la finalidad de que cada uno de los líderes de proyecto pueda llevar a cabo una evaluación de dichos proyectos durante el ciclo de vida de los mismos. Mediante el sistema propuesto cada líder de proyecto tendrá acceso a los resultados arrojados por cada una de las evaluaciones hecha al proyecto al que se encuentra asociado. El personal encargado por el Departamento Central de Calidad que tiene la responsabilidad de realizar esta tarea tendrá la posibilidad de acceder a todos y cada uno de los proyectos, así como a los resultados arrojados por la utilización de métricas durante el ciclo de vida de estos.

## ÍNDICE

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA</b> .....	<b>4</b>
<b>1.1 Introducción</b> .....	<b>4</b>
<b>1.2 Análisis de los principales modelos, metodologías y estándares para el desarrollo de software. Selección de la metodología que apoya la solución del problema.</b> .....	<b>4</b>
1.2.1 Rational Unified Process (RUP).....	4
1.2.2 Extreme Programing (XP). .....	5
1.2.3 Microsoft Solution Framework (MSF). .....	5
1.2.4 Selección de la metodología a utilizar. ....	6
<b>1.3 Tendencias y tecnologías actuales. Selección de las herramientas de desarrollo</b> .....	<b>7</b>
1.3.1 Principales lenguajes de programación. Selección del lenguaje a emplear en la implementación de la solución. ....	8
1.3.1.1 Practical Extracting and Reporting Language (PERL).....	8
1.3.1.2 Active Server Pages (ASP). ....	8
1.3.1.3 Hypertext Preprocessor (PHP).....	9
1.3.1.4 Fundamentación de la selección del lenguaje a utilizar. ....	10
1.3.2 Sistemas Gestores de Base de Datos (SGDB).....	10
1.3.2.1 PostgreSQL. ....	11
1.3.2.2 SQL Server. ....	11
1.3.2.3 MySQL. ....	12
1.3.2.4 Oracle.....	12
1.3.2.5 Fundamentación de la selección de la Base de Datos a utilizar. ....	13
1.3.3 CMS más utilizados en la actualidad. ....	13
1.3.3.1 CMS Drupal .....	13
1.3.3.2 CMS Joomla .....	15
1.3.3.3 CMS Plone .....	15
1.3.3.4 CMS E107.....	17
1.3.3.5 Fundamentación de la selección del CMS a utilizar. ....	18
<b>1.4 Conclusiones del Capítulo</b> .....	<b>19</b>
<b>CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA</b> .....	<b>20</b>
<b>2.1 Introducción</b> .....	<b>20</b>
<b>2.2 Breve exposición de la situación actual.</b> .....	<b>20</b>
<b>2.3 Propuesta de sistema</b> .....	<b>21</b>
2.3.1 Personas relacionadas con el sistema. ....	21
2.3.2 Requerimientos funcionales del sistema. ....	22
2.3.3 Requerimientos no funcionales del sistema. ....	23
<b>2.4 Conclusiones del Capítulo</b> .....	<b>24</b>
<b>CAPÍTULO 3. EXPLORACIÓN Y PLANIFICACIÓN</b> .....	<b>26</b>
<b>3.1 Introducción</b> .....	<b>26</b>

---

<b>3.2 Fase de exploración.</b> .....	<b>26</b>
3.2.1 Historias de Usuarios. ....	26
<b>3.3 Fase de planificación.</b> .....	<b>31</b>
3.3.1 Estimación de esfuerzo por historias de usuario. ....	32
3.3.2 Plan de Iteraciones.....	32
3.3.3 Plan de duración de las iteraciones. ....	33
3.3.4 Plan de entregas. ....	34
<b>3.4 Módulos Implementados.</b> .....	<b>36</b>
<b>3.5 Conclusiones del Capítulo.</b> .....	<b>39</b>
<b>CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA.</b> .....	<b>40</b>
<b>4.1 Introducción.</b> .....	<b>40</b>
<b>4.2 Fase de Implementación.</b> .....	<b>40</b>
4.2.1 Iteración 1 .....	40
4.2.2 Iteración 2 .....	49
4.2.3 Iteración 3 .....	56
<b>4.3 Pruebas</b> .....	<b>60</b>
<b>4.4 Pruebas de aceptación.</b> .....	<b>60</b>
<b>4.5 Conclusiones del Capítulo.</b> .....	<b>68</b>
<b>CAPITULO 5: ESTUDIO DE FACTIBILIDAD</b> .....	<b>69</b>
<b>5.1 Introducción.</b> .....	<b>69</b>
<b>5.2 Características del proyecto.</b> .....	<b>69</b>
<b>5.3 Cálculo de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.</b> .....	<b>71</b>
<b>5.4 Beneficios tangibles e intangibles.</b> .....	<b>74</b>
<b>5.5 Análisis de costo.</b> .....	<b>75</b>
<b>5.6 Conclusiones del Capítulo.</b> .....	<b>75</b>
<b>CONCLUSIONES GENERALES</b> .....	<b>76</b>
<b>RECOMENDACIONES.</b> .....	<b>77</b>
<b>BIBLIOGRAFÍA</b> .....	<b>78</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.</b> .....	<b>80</b>
<b>GLOSARIO DE TÉRMINOS.</b> .....	<b>82</b>

## ÍNDICE DE TABLAS

Tabla 1 Personas relacionadas con el sistema.....	22
Tabla 2 HU Autenticar Usuario.....	26
Tabla 3 HU Administrar Usuario.....	27
Tabla 4 HU Gestionar Proyecto.....	27
Tabla 5 HU Gestionar Categoría.....	28
Tabla 6 HU Gestionar Métrica.....	28
Tabla 7 HU Utilizar Métrica.....	29
Tabla 8 HU Obtener Reporte.....	29
Tabla 9 HU Asignar Proyecto.....	30
Tabla 10 HU Obtener Comportamiento por Métrica.....	30
Tabla 11 HU Obtener Historial de Proyecto por Métrica.....	31
Tabla 12 Estimación del esfuerzo por Historias de Usuario.....	32
Tabla 13 Plan de duración de las iteraciones.....	34
Tabla 14 Módulos y HU abarcadas.....	34
Tabla 15 Plan de duración entrega.....	35
Tabla 16 Tarjeta CRC Módulo proyecto.....	36
Tabla 17 Tarjeta CRC Módulo reporte.....	37
Tabla 18 Tarjeta CRC Módulo resultado.....	37
Tabla 19 Tarjeta CRC Módulo categoría.....	37
Tabla 20 Tarjeta CRC Módulo administrar.....	38
Tabla 21 Tarjeta CRC Módulo reporte_líder.....	38
Tabla 22 Tarjeta CRC Módulo métrica.....	38
Tabla 23 Módulos abordados en la primera iteración.....	41
Tabla 24 Tarea # 1 del módulo user.....	41
Tabla 25 Tarea # 2 del módulo user.....	41
Tabla 26 Tarea # 3 del módulo user.....	42
Tabla 27 Tarea # 4 del módulo user.....	42
Tabla 28 Tarea # 5 del módulo user.....	43
Tabla 29 Tarea # 6 del módulo user.....	43
Tabla 30 Tarea # 7 del módulo user.....	44
Tabla 31 Tarea # 8 del módulo user.....	44
Tabla 32 Tarea # 1 del módulo proyecto.....	45
Tabla 33 Tarea # 2 del módulo proyecto.....	45
Tabla 34 Tarea # 3 del módulo proyecto.....	46
Tabla 35 Tarea # 4 del módulo proyecto.....	46
Tabla 36 Tarea # 5 del módulo proyecto.....	46
Tabla 37 Tarea # 1 del módulo categoría.....	47
Tabla 38 Tarea # 2 del módulo categoría.....	47
Tabla 39 Tarea # 3 del módulo categoría.....	48
Tabla 40 Tarea # 4 del módulo categoría.....	48
Tabla 41 Tarea # 5 del módulo categoría.....	49
Tabla 42 Módulos abordados en la segunda iteración.....	49
Tabla 43 Tarea # 1 del módulo métrica.....	50
Tabla 44 Tarea # 2 del módulo métrica.....	50
Tabla 45 Tarea # 3 del módulo métrica.....	51

<i>Tabla 46 Tarea # 4 del módulo métrica.....</i>	<i>51</i>
<i>Tabla 47 Tarea # 5 del módulo métrica.....</i>	<i>52</i>
<i>Tabla 48 Tarea # 1 del módulo resultado.....</i>	<i>52</i>
<i>Tabla 49 Tarea # 2 del módulo resultado.....</i>	<i>53</i>
<i>Tabla 50 Tarea # 3 del módulo resultado.....</i>	<i>53</i>
<i>Tabla 51 Tarea # 3 del módulo resultado.....</i>	<i>54</i>
<i>Tabla 52 Tarea # 1 del módulo resultado.....</i>	<i>54</i>
<i>Tabla 53 Tarea # 2 del módulo resultado.....</i>	<i>55</i>
<i>Tabla 54 Tarea # 3 del módulo resultado.....</i>	<i>55</i>
<i>Tabla 55 Módulos abordados en la tercera iteración.....</i>	<i>56</i>
<i>Tabla 56 Tarea # 1 del módulo administrar.....</i>	<i>56</i>
<i>Tabla 57 Tarea # 2 del módulo administrar.....</i>	<i>57</i>
<i>Tabla 58 Tarea # 1 del módulo reporte.....</i>	<i>57</i>
<i>Tabla 59 Tarea # 2 del módulo reporte.....</i>	<i>58</i>
<i>Tabla 60 Tarea # 3 del módulo reporte.....</i>	<i>58</i>
<i>Tabla 61 Tarea # 1 del módulo reporte_lider.....</i>	<i>59</i>
<i>Tabla 62 Tarea # 2 del módulo reporte_lider.....</i>	<i>59</i>
<i>Tabla 63 Tarea # 3 del módulo reporte_lider.....</i>	<i>60</i>
<i>Tabla 64 Prueba de aceptación para la historia de usuario Autenticar Usuario.....</i>	<i>61</i>
<i>Tabla 65 Prueba de aceptación para la historia de usuario Administrar Usuario.....</i>	<i>61</i>
<i>Tabla 66 Prueba de aceptación para la historia de usuario Gestionar Proyecto.....</i>	<i>62</i>
<i>Tabla 67 Prueba de aceptación para la historia de usuario Gestionar Proyecto.....</i>	<i>62</i>
<i>Tabla 68 Prueba de aceptación para la historia de usuario Gestionar Proyecto.....</i>	<i>63</i>
<i>Tabla 69 Prueba de aceptación para la historia de usuario Gestionar Categoría.....</i>	<i>63</i>
<i>Tabla 70 Prueba de aceptación para la historia de usuario Gestionar Categoría.....</i>	<i>63</i>
<i>Tabla 71 Prueba de aceptación para la historia de usuario Gestionar Categoría.....</i>	<i>64</i>
<i>Tabla 72 Prueba de aceptación para la historia de usuario Gestionar Métrica.....</i>	<i>64</i>
<i>Tabla 73 Prueba de aceptación para la historia de usuario Gestionar Métrica.....</i>	<i>65</i>
<i>Tabla 74 Prueba de aceptación para la historia de usuario Gestionar Métrica.....</i>	<i>65</i>
<i>Tabla 75 Prueba de aceptación para la historia de usuario Utilizar Métrica.....</i>	<i>66</i>
<i>Tabla 76 Prueba de aceptación para la historia de usuario Obtener Reporte.....</i>	<i>66</i>
<i>Tabla 77 Prueba de aceptación para la historia de usuario Asignar Proyecto.....</i>	<i>66</i>
<i>Tabla 78 Prueba de aceptación para la historia de usuario Obtener Comportamiento por Métrica.....</i>	<i>67</i>
<i>Tabla 79 Prueba de aceptación para la historia de usuario Obtener Historial de proyecto por Métrica.....</i>	<i>67</i>
<i>Tabla 80 Entradas externas.....</i>	<i>69</i>
<i>Tabla 81 Salidas externas.....</i>	<i>69</i>
<i>Tabla 82 Consultas Externas.....</i>	<i>69</i>
<i>Tabla 83 Archivos lógicos internos.....</i>	<i>70</i>
<i>Tabla 84 Puntos de función desajustados.....</i>	<i>71</i>
<i>Tabla 85 Características.....</i>	<i>72</i>
<i>Tabla 86 Factores de escala.....</i>	<i>72</i>
<i>Tabla 87 Multiplicadores de esfuerzo.....</i>	<i>72</i>
<i>Tabla 88 Resultados.....</i>	<i>74</i>

## INTRODUCCIÓN

Uno de los problemas que se afrontan actualmente en la esfera de la informática es la calidad del software. Para obtener un software con calidad es necesario la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, con el objetivo de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

Una manera eficaz para evaluar algunos parámetros como funcionalidad, complejidad y eficiencia durante el proceso de producción de software es la utilización de métricas, debido a que estas permiten evaluar la productividad de los desarrolladores, establecer una línea base para la estimación y de igual forma nos ayuda a justificar el uso de nuevas herramientas.

Uno de los pilares fundamentales de la Universidad de las Ciencias Informáticas (UCI), es la producción de productos de software de alta calidad a través de los proyectos productivos, donde un gran grupo de personas se dedica a comprobar la calidad y la eficiencia tanto de los productos de software como de los procesos productivos, sin embargo no se posee un espacio donde los directivos del Departamento de Calidad de la Universidad puedan brindar a los jefes de proyecto una manera fácil y didáctica de verificar el rendimiento de su proyecto a través del uso de un conjunto de métricas, además no se cuenta con un registro periódico de los datos arrojados en la aplicación de dichas métricas, para una futura estimación o investigación.

Esta propuesta surge debido a que actualmente en la producción de software existen pocas herramientas para la medición sistemática de la calidad durante el proceso de desarrollo de software mediante la utilización de métricas, tal es el caso de la herramienta Scope que cumple las funcionalidades básicas que el sistema debe tener; ya que puede hacer una estimación de costo, recursos y duración del proyecto así como reunir datos de este para una evaluación comparativa, pero tiene la desventaja de ser un software propietario.

Basado en lo anteriormente expuesto es que la dirección de Calidad de la Universidad decidió hacer un software que respondiera íntegramente a las necesidades de la UCI, ya que cuenta con el potencial necesario para la realización del mismo, mediante el uso de herramientas y tecnologías para obtener un producto totalmente libre.

De ahí se puede establecer que el **problema** a solucionar consiste en: ¿Cómo contribuir a la estimación, el seguimiento y el control de los procesos productivos de software en la Universidad de las Ciencias Informáticas?

Por tanto el **objeto de estudio** de este trabajo es la utilización de métricas para evaluar la eficiencia y calidad en el proceso de producción de Software.

De ello se deriva que el **campo de acción** que abarca este trabajo, es el proceso de aplicación de métricas mediante las cuales será comprobada la eficiencia y calidad durante el proceso de producción de Software en la Universidad de las Ciencias Informáticas.

De ahí que nuestra **idea a defender** sea que si se desarrolla un sistema informático que sea capaz de gestionar y controlar métricas de calidad, será posible llevar a cabo una inspección sistemática de los proyectos productivos de la UCI, mejorando en gran medida la eficiencia de los mismos.

El **objetivo general** de este trabajo, será desarrollar un sistema automatizado que permita gestionar las métricas que serán utilizadas como herramienta para evaluar la eficiencia y calidad en el proceso de producción de Software en la Universidad de las Ciencias Informáticas.

Los **objetivos específicos** son:

- ❖ Realizar un estudio del arte asociado al problema.
- ❖ Realizar análisis y diseño del sistema.
- ❖ Desarrollar el sistema de gestión de métricas propuesto.

Para cumplir con estos objetivos y resolver la situación problemática planteada, se proponen las siguientes **Tareas**:

- ❖ Investigación sobre la utilización de métricas durante el proceso productivo de Software.
- ❖ Análisis y selección de las tecnologías y herramientas a emplear para el desarrollo del sistema.
- ❖ Selección de la Metodología de Análisis y Diseño de Sistemas Informáticos, que faciliten la creación y garanticen la calidad del sistema.
- ❖ Diseño de una Base de datos que soporte las funcionalidades del sistema.
- ❖ Realizar el análisis y diseño del sistema que se propone.
- ❖ Implementación del sistema propuesto.

El trabajo está estructurado en: cinco capítulos, conclusiones, recomendaciones, bibliografía y glosario.

El **Capítulo 1** presenta un estudio de las tendencias actuales de desarrollo de software con el propósito de realizar una selección de las herramientas más eficientes en el desarrollo de la aplicación que dará solución al problema.

Para lograr los objetivos de esta fase, se expone un análisis de las metodologías de desarrollo de software más comúnmente utilizadas, así como las características esenciales de los principales lenguajes de programación empleados en el desarrollo de aplicaciones Web y en ese mismo sentido se analizan las particularidades y funcionalidades de los posibles sistemas gestores de bases de datos a emplear en la solución del problema propuesto.

En el **Capítulo 2** se realiza la descripción de la situación actual referente a la evaluación y seguimiento de los proyectos productivos mediante la utilización de métricas de calidad en la UCI. Se realiza además la propuesta de las características que presentará el sistema informático que dará respuesta a la problemática existente donde se interrelacionan los procesos involucrados en la estrategia con las funcionalidades que permitirá realizar el mismo. Por otra parte se confecciona el modelo de dominio que agrupa los conceptos fundamentales en función de una mejor captura de requisitos. Se definen los requisitos funcionales y no funcionales y se confecciona un diagrama de casos de uso, donde se identifican las relaciones de los actores con el sistema.

En el **Capítulo 3** se hace énfasis en las fases de exploración y planificación propias de la metodología de desarrollo utilizadas para la implementación del sistema que se propone. Se presentan además los artefactos generados durante el transcurso de las mismas.

En el **Capítulo 4** se detallan las tres iteraciones llevadas a cabo durante el desarrollo del sistema, exponiendo las tareas generadas al analizar cada una de las historias de usuario, así como las pruebas de aceptación efectuadas sobre el sistema.

En el **Capítulo 5** se realiza un análisis del costo, el esfuerzo y los beneficios es de suma importancia a la hora de implementar un sistema. En este capítulo se detalla el método estimación aplicado al sistema propuesto.

## **CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA**

### **1.1 Introducción**

En este capítulo se realiza un estudio de las tendencias actuales de desarrollo de software con el propósito de realizar una selección de las herramientas más eficientes en el desarrollo de la aplicación que dará solución al problema.

Para lograr los objetivos de esta fase, se expone un análisis de las metodologías de desarrollo de software más comúnmente utilizadas, así como las características esenciales de los principales lenguajes de programación empleados en el desarrollo de aplicaciones Web y en ese mismo sentido se analizan las particularidades y funcionalidades de los posibles sistemas gestores de bases de datos a emplear en la solución del problema propuesto.

### **1.2 Análisis de los principales modelos, metodologías y estándares para el desarrollo de software. Selección de la metodología que apoya la solución del problema.**

A nivel internacional las instituciones y empresas dedicadas a la industria del software emplean en su actividad de desarrollo, modelos, metodologías o procedimientos estándares para desarrollar, instalar y mantener un producto de este tipo. Dichos modelos permiten uniformar la filosofía de trabajo, en aras de lograr la obtención de un software de calidad, y a la vez elevar la productividad para así competir y poder acceder a las oportunidades del mercado mundial.

Para lograr un correcto desarrollo de software, estas metodologías establecen un conjunto de actividades que definen cómo se debe hacer el software, quién debe hacer cada actividad, cuándo hacerla y qué se debe hacer.

En la actualidad existe gran cantidad de metodologías orientadas al proceso de desarrollo de software, entre las más significativas se pueden mencionar el *Rational Unified Process (RUP)*, *Extreme Programming (XP)* y *Microsoft Solution Framework (MSF)*.

#### **1.2.1 Rational Unified Process (RUP).**

El *Rational Unified Process (RUP)* es un proceso que define quién, qué, cómo y cuándo deben hacerse las cosas. Está fundamentada en un enfoque orientado a modelos de desarrollo basado en componentes, utilizando para ello el Lenguaje de Modelado Unificado (*UML, Unified Modeling Language*) el que define técnicas de análisis y diseño que ayudan a la confección de una solución sólida de software.

RUP se caracteriza por ser **Dirigido por casos de uso** donde los casos de uso definen lo que el usuario desea a partir de la captura de requisitos y la modelación del negocio. Es **Centrado en la Arquitectura**, característica que brinda una visión completa del sistema, se describen los procesos del negocio que son más importantes, para comprenderlo, desarrollarlo y producirlo de una forma eficaz. **Iterativo e Incremental** donde cada fase se desarrolla en iteraciones, de forma tal que se pueda dividir en pequeños proyectos mejorando su comprensión y desarrollo.

De forma general RUP se divide en 4 fases el desarrollo del software, estas son: **inicio** en la que se define el modelo del negocio, el alcance y los límites del proyecto, **elaboración** en la cual se define, valida y cimienta la arquitectura, **construcción** donde se desarrolla el producto y **transición** es la encargada de poner el producto en manos de los usuarios.

### **1.2.2 Extreme Programming (XP).**

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizadas para proyectos que consten de un pequeño equipo de trabajo y cuyo plazo de entrega es como máximo de dos a tres meses. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. [11]

Esta metodología se basa en tres características fundamentales, las **Pruebas Unitarias** que son un conjunto de pruebas realizadas a los principales procesos, adelantándonos de esta forma a posibles fallos que pudieran ocurrir en el futuro. **La Refabricación** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio y por otra parte se encuentra la **Programación en Pares** que no es más que la participación de dos desarrolladores en un proyecto en la misma estación de trabajo. [11]

Su objetivo principal esta en lograr la completa satisfacción del cliente, proporcionándole lo que necesita y cuando lo necesita, además de potenciar al máximo el trabajo en grupo. Dicha metodología solo puede funcionar con programadores experimentados en la materia, y solo para pequeños grupos de estos. [11]

### **1.2.3 Microsoft Solution Framework (MSF).**

*Microsoft Solution Framework* es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos. Se centra en el modelo de procesos y de equipo dejando en un segundo plano las elecciones tecnológicas. [11]

MSF Organiza los procesos necesarios para crear y entregar un proyecto, dividiendo el desarrollo de este en cinco fases: **Visión**, donde se hace una descripción general de las metas y restricciones del proyecto. Aquí se identifican las tareas y entregables. **Planeamiento**, los miembros del equipo de trabajo y el cliente definen el que y el cómo de la solución a implementar. **Desarrollo**, en esta fase los miembros del equipo de trabajo desarrollan y prueban la solución a implementar, este punto comprende el desarrollo de código y los entregables de documentación resultantes. **Estabilización**, es en la que los miembros del equipo de trabajo y del cliente prueban la solución completa, estabilizando la misma en función de los desvíos encontrados y preparando la misma para su liberación en producción. **Implementación**, donde el equipo implementa la solución tecnológica y sus componentes, estabiliza la implementación, transfiere el proyecto a producción y soporte, y obtiene la aprobación final del cliente sobre el proyecto. [11]

Esta metodología se puede utilizar para pequeños y grandes proyectos, es un proceso que se basa en la colaboración entre todos los que realizan el proyecto, analiza riesgos y provee de plantillas que nos ayudan a la hora de la documentación.

A pesar de que nos ayuda en la documentación esta es muy grande por lo que se necesita una gran paciencia si se trabaja con este proceso. Además como es Microsoft su campo de trabajo solo se limita a usar herramientas de este mismo proveedor.

#### ***1.2.4 Selección de la metodología a utilizar.***

En correspondencia con el análisis realizado anteriormente sobre las metodologías seleccionadas, se puede evidenciar la presencia de diferentes características para cada una de ellas, y la particularidad de poder ser empleadas para diferentes situaciones respectivamente.

RUP es perfectamente idónea para proyectos de largo plazo, aunque es caracterizada por ser una metodología compleja. MSF por su parte es una metodología adaptable a proyectos de cualquier dimensión, sin embargo, no basa su funcionamiento en la documentación del software sino en el control de entregables por parte de los desarrolladores.

La metodología **XP** está destinada a proyectos de corto plazo, se basa en iteraciones pequeñas y está dirigida fundamentalmente a los clientes.

Teniendo en cuenta las particularidades de cada una, se ha seleccionado como apoyo en el desarrollo de la aplicación propuesta la metodología XP, debido a que se adapta en gran medida tanto al tipo de proyecto a desarrollar como a las condiciones de trabajo, el proyecto es pequeño, los requisitos del sistema cambian frecuentemente, el riesgo de desarrollo es elevado debido al corto tiempo de entrega planteado y a los continuos cambios de requerimientos. Las metodologías

tradicionales imponen un proceso disciplinado para tratar de hacer el trabajo predecible, eficiente y planificado. XP es más liviana y ágil y está orientada más a las personas que a los procesos.

### 1.3 Tendencias y tecnologías actuales. Selección de las herramientas de desarrollo.

Para dar solución a la problemática planteada se ha propuesto desarrollar una Aplicación Web, según la petición realizada por los clientes, teniendo en cuenta las grandes funcionalidades y posibilidades de implementación que brindan las herramientas de desarrollo en esta área.

Este tipo de aplicaciones presenta como principal peculiaridad el empleo del modelo cliente/servidor, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

En este modelo, las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

El programa cliente gestiona la comunicación con el servidor y ofrece las herramientas necesarias para poder trabajar con dicho servidor Web. [14]

El servidor Web es un pilar importante en esta tecnología, puesto que es el programa que corre en una máquina central escuchando las peticiones de los clientes y respondiendo a las mismas en forma de una página que es interpretada por éste y mostrada al usuario de una manera entendible para él. Los más conocidos y utilizados en la actualidad son Apache y el IIS (*Internet Information Server*).

El IIS es un software privativo que trabaja solo sobre Windows, por lo que en la solución propuesta se empleará el **Apache**, éste por su parte es un servidor que presenta un sin número de características que lo hacen estar entre los primeros en este ámbito. Es flexible, rápido y eficiente, puede ser ejecutado en cualquier plataforma y adaptado a diferentes entornos y necesidades. Se desarrolla de forma abierta y gracias a que es modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor (como se expondrá más adelante).

Para poder seleccionar las herramientas a emplear en la implementación de la aplicación Web, se realizará un estudio de las principales tecnologías, lenguajes de programación y gestores de bases de datos.

### **1.3.1 Principales lenguajes de programación. Selección del lenguaje a emplear en la implementación de la solución.**

En la actualidad existen varios lenguajes de programación que permiten desarrollar una aplicación Web, los cuales se dividen en dos grupos fundamentales que reconoce la arquitectura Cliente/Servidor: Los lenguajes del lado del Servidor y los lenguajes del lado del Cliente.

Entre los lenguajes del lado del servidor más comúnmente utilizados por los programadores podemos encontrar: el *Personal Home Page* (PHP), *Practical Extracting and Reporting Language* (PERL), *Active Server Pages* (ASP). Estos lenguajes se caracterizan de forma general por desarrollar la lógica del negocio dentro del Servidor, de realizar el acceso y la actualización a Base de Datos, así como del tratamiento de la información.

Del lado del Cliente tenemos: *JavaScript* y al *Visual Basic Script*, que son los encargados de aportar dinamismo a la aplicación en los navegadores permitiendo una mayor flexibilidad en la interfaz.

#### **1.3.1.1 Practical Extracting and Reporting Language (PERL).**

Perl es un lenguaje de programación de scripts que está disponible en múltiples plataformas y sistemas operativos. El desarrollo de aplicaciones en este lenguaje es muy rápido y se pueden incorporar a cualquier conjunto de instrucciones de él, una colección enorme de módulos que están favorables para su utilización. [12]

Perl es "Software Libre", lo que quiere decir que el código fuente está disponible para que cualquiera lo pueda ver, utilizar o modificar, brindándole de esta forma al programador mucha libertad para que haga el programa como desee. [12]

Perl trae consigo una serie de módulos DBD (Controlador de base de datos) que manejan los detalles de acceso a diferentes bases de datos y usa a menudo como un "lenguaje pegamento", ligando sistemas e interfaces que no fueron diseñados específicamente para interoperar. [12]

Puede ser más lento que otros lenguajes haciendo lo mismo porque tiene que compilar el código fuente cada vez que corre el programa ya que no puede salvar su compilación intermedia como lo hacen Java y *Python*, los scripts Perl conllevan esta sobrecarga en cada ejecución utilizando muchos recursos de la máquina, lo que provoca que no sea tan ligero como otros lenguajes. [12]

#### **1.3.1.2 Active Server Pages (ASP).**

Es un conjunto de tecnologías de desarrollo de aplicaciones Web comercializado por Microsoft. Es usado por programadores para construir sitios Web domésticos, aplicaciones Web y servicios XML. Forma parte de la plataforma NET de Microsoft, de lo cual se deduce que es un lenguaje privativo y

de un alto costo. Este soporta acceso a bases de datos, trabajo con archivos y carpetas, envío de email, paginación de resultados, procesado de formularios, y muchas otras opciones. [5]

En una página ASP podemos incluir casi todo: HTML plano, código de scripting, texto. No hay una distinción formal entre el contenido de una página y su comportamiento. Facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente, hasta la implementación y la configuración de sitios. [5]

Además ASP es un sistema con nula portabilidad pues requiere necesariamente de un servidor Windows, con todas las implicaciones de alto costo, poca flexibilidad y escasa seguridad que estos equipos brindan.

### **1.3.1.3 Hypertext Preprocessor (PHP)**

PHP (siglas que originalmente significaban *Personal Home Page*), es un lenguaje script para el desarrollo de páginas Web dinámicas del lado del servidor. Es *Open Source* (código abierto), interpretado y de alto nivel, especialmente pensado para desarrollos Web. [4]

Es un lenguaje robusto y estable donde se pueden hacer grandes cosas con pocas líneas de código, por lo que es útil su utilización. El código de este es mucho más legible que el de PERL y muy sencillo de aprender, además de que viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor. [4]

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos. Este lenguaje es multiplataforma, tiene la capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL, aunque puede interactuar también con motores de bases de datos tales como MS SQL, Oracle, Informix, PostgreSQL, y otros muchos, con un excelente soporte. [15]

PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz y está completamente escrito en C, por lo que se ejecuta rápidamente utilizando poca memoria. [4]

Este lenguaje soporta en cierta medida la programación orientada a objeto (Clases y herencia) y también realiza de forma automática el análisis léxico para recoger las variables que se pasan en la dirección, lo cual libra al usuario de tener que separar las variables y sus valores. La comprobación de que los parámetros son válidos se hace en el servidor y no en el cliente, de forma que se puede evitar que se reciban solicitudes adulteradas. Además PHP viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos. [15]

Sin embargo, todo el trabajo lo realiza el servidor y no delega al cliente, por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número y dicha orientación a objetos es aún muy deficiente para aplicaciones grandes.

#### **1.3.1.4 Fundamentación de la selección del lenguaje a utilizar.**

Perl es un lenguaje de programación Web libre que brinda un sin número de características interesantes, sin embargo, consume muchos recursos de la máquina y se torna lento en aplicaciones de bajo nivel. ASP por su parte es privativo lo que justifica su enorme precio. PHP pertenece al grupo de programas de código abierto y libre.

Se ha seleccionado como lenguaje a utilizar en la implementación de la aplicación Web, al **PHP**. Este es un lenguaje gratuito, multiplataforma y debido a sus características hace posible que el cliente interactúe con una página rápida, eficiente y segura, capaz de mostrar y procesar información.

Unido a lo anterior, se puede mencionar que es muy sencillo y legible, y cuenta con una amplia librería de funciones que permiten hacer cualquier tipo de operación, como trabajo con archivos y carpetas, procesamiento de formularios, paginación de resultados y muchas otras opciones, lo cual concede la posibilidad de realizar un producto de acuerdo a las necesidades del usuario.

#### **1.3.2 Sistemas Gestores de Base de Datos (SGDB).**

Los Sistemas Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. En las bibliografías de este tema, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y Data Base Management Sistema, según su expresión inglesa.

El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado. Un SGBD permite: la Independencia entre los datos y los programas de aplicación, minimizar las redundancias, integrar y sincronizar las bases de datos, garantiza la integridad, la seguridad y protección de los datos, facilita la de manipulación de la información y permite un control centralizado.

Entre los SGBD más comunes se encuentra el Oracle, el Microsoft SQL Server, el MySQL y el PostgreSQL, a continuación se expondrán las características fundamentales de cada uno de ellos, lo que permitirá la selección del que se va a utilizar en la implementación de la Aplicación Web.

### **1.3.2.1 PostgreSQL.**

PostgreSQL es el servidor de bases de datos de código abierto más potente que existe, es además la alternativa más cercana a MySQL cuando se precisa de operaciones avanzadas como transacciones, procedimientos almacenados, vistas, o cuando se precisa de una base de datos que soporte gran cantidad de información. [2]

Es el servidor más utilizado por todos aquellos programadores que realizan aplicaciones cliente servidor, complejas o críticas.

Es muy utilizado actualmente y es una alternativa económica a SQL Server, pues su costo es menor y tiene similares prestaciones. Este se puede utilizar sobre cualquier sistema operativo, característica que lo pone por encima de SQL Server y al parejo con MySQL. [9]

La mayor limitación que posee viene dada por su velocidad ya que se torna un tanto lento, en la realización de operaciones elementales de los sistemas gestores de base de datos. [2]

Además consume muchos recursos por lo que puede sobrecargar el sistema, tiene un límite de tamaño para los registros, y aunque es compatible con PHP tiene muy pocos comandos definidos para el tratamiento de este. [9]

### **1.3.2.2 SQL Server.**

SQL Server es un sistema de base de datos muy completo y potente. Posee una gran velocidad y soporta un volumen de datos muy grande. También, presenta mecanismos que le permiten realizar sentencias complicadas, lo que lo hace perfectamente adecuado para aplicaciones críticas y con cualquier grado de complejidad. Por otro lado reserva una parte de la base de datos para guardar el registro de transacciones con los comandos pendientes, lo que asegura que independientemente de que el programador use o no transacciones en su código, en ningún caso la base de datos quedaría en un estado inconsistente debido a una ejecución parcial de comandos. Además ofrece características avanzadas orientadas a la integridad de la base de datos. [13]

Podemos decir que ayuda a desarrollar e implantar aplicaciones empresariales más escalables, fiables y seguras optimizando la productividad del sector TI reduciendo la complejidad en la creación, implantación y administración de las aplicaciones de bases de datos. Aumenta las capacidades de los desarrolladores con un entorno de desarrollo valioso, flexible y actual para que creen bases de datos más seguras además de compartir datos a través de múltiples aplicaciones y dispositivos para facilitar la interconexión entre sistemas internos y externos. [13]

Posee el inconveniente que corre solamente sobre una plataforma, Windows, además de ser un software privativo con un alto costo.

### **1.3.2.3 MySQL.**

MySQL por su parte es un servidor de base de datos que ostenta la cualidad de poseer un rápido nivel de procesamiento. Es el más utilizado cuando se emplean lenguajes de programación como PHP y Perl, y cabe mencionar su gran afinación principalmente con PHP.

MySQL es un software libre, confiable y fácil de usar, es multiplataforma, multiusuario y permite elaborar consultas con el robusto SQL, consume muy pocos recursos tanto del CPU como de memoria, presenta mejoras en utilidades de administración (backup, recuperación de errores, etc.), no hay límites en el tamaño de los registros, tiene como una de sus principales ventajas la velocidad en la lectura de datos.

Es un servidor multi-hilos de bases de datos de código abierto confiable, multiproceso, compacto y poderoso donde podemos hacer las bases de datos a código abierto, el principal objetivo de MySQL es velocidad y robustez, escrito en C y C++ y posee un sistema de contraseñas y privilegios muy flexible y seguro.[7]

En recientes versiones de MySQL se incluyen un conjunto de facilidades que no tenía en versiones anteriores como el uso de procedimientos almacenados, bloqueo de registros, integridad referencial, transacciones, entre otros, lo que aumenta aún más sus ventajas. (Schumacher, 2004)

### **1.3.2.4 Oracle.**

Es considerado el SGBD más completo que existe. Sus características más destacadas son el soporte de transacciones, su gran estabilidad y seguridad, su escalabilidad, así como que es un sistema multiplataforma, entre otras ventajas. Su elevado precio hace que sólo se vea en empresas muy grandes y multinacionales en forma general. En el desarrollo de páginas Web pasa lo mismo ya que como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, Access, MySQL, SQL Server, etc. [7]

Un aspecto que ha sido criticado por algunos especialistas es la seguridad de la plataforma, y las políticas de suministros de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios. [7]

Hasta hace poco su dominio en el mercado de los servidores de bases de datos empresariales era casi total, pero recientemente está sufriendo la competencia del SQL Server de Microsoft y de la oferta de otros SGBD libres como MySQL y Postgre. [7]

### **1.3.2.5 Fundamentación de la selección de la Base de Datos a utilizar.**

Después de haber realizado un análisis completo de los diferentes tipos de bases de datos más utilizados en la actualidad y teniendo en cuenta los aspectos más significativos por los cuales se puede comparar, se llega a la conclusión de que el más idóneo para llevar a cabo los objetivos es

#### **MySql.**

Poniendo especial atención a las restricciones del cliente, es un software libre y además es utilizable en cualquier plataforma. Por otro lado y comparándolo con los otros tipos de bases de datos su mayor adversario es SQL Server el cual es software propietario. Postgre por su parte a pesar de su robustez se torna un poco lento. Además MySql posee una gran afinidad con PHP y es el más eficiente para aplicaciones no tan complejas como la que se pretende desarrollar.

### **1.3.3 CMS más utilizados en la actualidad.**

En la actualidad es muy importante que se le preste la merecida importancia a los CMS (Content Management System). Los CMS son aplicaciones altamente configurables que brindan la posibilidad de manipular contenidos de propósito general, aunque se pueden personalizar todo lo que se quiera. El principal objetivo de los CMS es proveer al desarrollador de una herramienta para la construcción de aplicaciones Web que manipulen contenidos de forma dinámica.

La utilización de CMS simplifica bastante las tareas de mantenimiento de las páginas Web, puesto que al estar separada la presentación del contenido, podemos modificar los ficheros de presentación para cambiar el aspecto, e incluso la arquitectura del sitio, y seguir teniendo los mismos contenidos y accediendo a ellos sin problema. Estos Sistemas de Gestión de Contenidos tienen varias funcionalidades: la creación de contenido, gestión de contenido, publicación y presentación. A continuación se presenta una comparación entre algunos de los CMS más utilizados en la actualidad de forma tal que se puedan identificar las características fundamentales de cada uno de ellos:

#### **1.3.3.1 CMS Drupal**

Drupal es un CMS para sitios Web poderoso, conocido por la calidad de su código y por la seguridad que ofrece. Es un sistema estable, de actualización continua y dinámica debido a que en lugar de almacenar sus contenidos en archivos estáticos, en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos.

Drupal es basado en el lenguaje de programación PHP y base de datos MySQL. Es un sistema de fácil instalación y utilización, que almacena en la base de datos el contenido textual de las páginas del sitio y otras configuraciones. Posee una potente documentación que le permite al usuario una buena comprensión.

### **Requisitos del sistema**

- ❖ Servidor de aplicación: PHP.
- ❖ Gestor de Base de Datos: MySQL o PostgreSQL.
- ❖ Licencia: GPL.
- ❖ Lenguaje de Programación: PHP.
- ❖ Servidor Web: Apache, IIS.

### **Apoyo**

- ❖ Posee manuales comerciales.
- ❖ Posee Foro Público.
- ❖ Ayuda En línea.
- ❖ Apoyo comercial.

### **Facilidad de Empleo**

- ❖ Posee URL amistoso.
- ❖ Se puede subir archivos.
- ❖ Editor WYSIWYG.
- ❖ Se puede añadir verificador de ortografía.

### **Administración**

- ❖ Posee administración en línea.
- ❖ Cambio de estilo y plantilla de la Web.
- ❖ Traducción de la Web.

### **Usos**

- ❖ Posee blog.
- ❖ Foro de Discusión.
- ❖ Calendario.
- ❖ Motor De búsqueda.
- ❖ Encuestas.
- ❖ Gráficos.

- ❖ Se puede añadir libro de invitados.

### 1.3.3.2 CMS Joomla

Joomla es un Sistema de Gestión de Contenido de código abierto desarrollado bajo la licencia GPL que permite crear sitios Web dinámicos e interactivos. Tiene como objetivo principal dar soluciones a las necesidades de todos aquellos que participan de alguna manera en un proyecto.

Joomla está construido con PHP, es de fácil instalación y uso. Este gestor de contenidos se usa para hacer publicaciones en Internet e intranets utilizando una base de datos MySQL. Actualmente los programadores han publicado Joomla 1.5b bajo un código completamente reescrito y construido bajo PHP 5.

#### Requisitos del sistema

- ❖ Servidor de aplicación: Apache recomendado, pero se puede utilizar cualquier servidor con soporte para PHP.
- ❖ Gestor de Base de Datos: MySQL.
- ❖ Licencia: GPL.
- ❖ Lenguaje de Programación: PHP.
- ❖ Servidor Web: Apache.

#### Apoyo

- ❖ Posee manuales comerciales.
- ❖ Posee Foro Público.
- ❖ Ayuda en línea.
- ❖ Posee apoyo comercial.

### 1.3.3.3 CMS Plone

Plone es capaz de manejar las funciones básicas de un CMS ligero, como Drupal o Joomla, pero a diferencia de estos está mejor equipado para su uso en grandes entornos corporativos con grandes números de usuarios donde la información se maneja de forma personalizada. Entre sus funciones esenciales se encuentra un sólido motor de flujos de trabajo, tecnología Wiki incluida y un potente motor de búsquedas. Es una herramienta que se utiliza para potenciar procesos que siempre realizamos: organizar, buscar, entregar, adquirir, guardar, relacionar información, publicar, etc.

### **Requisitos del sistema**

- ❖ Servidor de aplicación: Zope.
- ❖ Gestor de Base de Datos: Zope.
- ❖ Licencia: GPL.
- ❖ Lenguaje de Programación: Python.
- ❖ Servidor Web: Apache, IIS, Zope.

### **Apoyo**

- ❖ Posee manuales comerciales.
- ❖ Posee Foro Público.
- ❖ Ayuda en línea limitada.
- ❖ Apoyo comercial.

### **Facilidad de Empleo**

- ❖ Posee URL amistoso.
- ❖ Se pueden subir archivos.
- ❖ Editor WYSIWYG.
- ❖ Se puede añadir verificador de ortografía.

### **Administración**

- ❖ Posee administración en línea.
- ❖ Cambio de estilo y plantilla de la Web.
- ❖ Traducción de la Web.

### **Usos**

- ❖ Posee blog.
- ❖ Foro de Discusión.

- ❖ Calendario.
- ❖ Motor de búsqueda.
- ❖ Gráficos.
- ❖ Se pueden añadir encuestas.
- ❖ Se puede añadir libro de invitados.

#### 1.3.3.4 CMS E107

**E107** es un Sistema de Manipulación de Contenidos (CMS) escrito en PHP, que usa MySQL como base de datos. Es completamente gratuito y está en constante desarrollo.

Se ha hecho un espacio importante en el mundo de los CMS de libre distribución. No es de los más conocidos, pero incorpora una serie de características muy depuradas a la hora de la administración de contenidos que lo convierten en uno de los favoritos en muchos lugares del mundo.

##### **Requisitos del sistema**

- ❖ Servidor de aplicación: Apache, IIS.
- ❖ Gestor de Base de Datos: MySQL.
- ❖ Licencia: GPL.
- ❖ Lenguaje de Programación: PHP, Javascript, XML, XHTML.
- ❖ Servidor Web: Apache, IIS.

##### **Apoyo**

- ❖ No posee manuales comerciales.
- ❖ Posee Foro Público.
- ❖ Ayuda en línea.
- ❖ No posee apoyo comercial.

##### **Facilidad de Empleo**

- ❖ Posee URL amistoso.
- ❖ No se pueden subir archivos.
- ❖ Editor WYSIWYG.

- ❖ No posee verificador de ortografía.

### **Administración**

- ❖ Posee administración en línea.
- ❖ Cambio de estilo y plantilla de la Web.
- ❖ Traducción de la Web.

### **Usos**

- ❖ Posee blog.
- ❖ Foro de Discusión.
- ❖ Calendario.
- ❖ Motor de búsqueda.
- ❖ Encuestas.
- ❖ No se pueden añadir gráficos.
- ❖ Libro de invitados.

#### **1.3.3.5 Fundamentación de la selección del CMS a utilizar.**

Ninguno de estos sistemas de gestión de contenidos se destaca por encima del otro, cada uno tiene sus ventajas e inconvenientes y es más o menos adecuado para cada sitio particular, en dependencia de de las necesidades concretas que dicho sitio requiera. Después de comparados y evaluados los CMS de código abierto previamente vistos y teniendo en cuenta las características de cada uno de ellos y las exigencias del software a implementar, se ha seleccionado el sistema de gestión de contenidos Drupal pues aunque su puesta en marcha resulte un poco más complicada que algunos de los demás, la integración simplifica notablemente su gestión, posee un buen diseño y un código de muy buena calidad, que lo hace fácilmente extensible lo cual lo convierte en una base en la que se puede integrar todo tipo de módulos. La usabilidad y accesibilidad son excelentes en comparación con los CMS antes mencionados además Drupal genera un código HTML extremadamente limpio y bien estructurado y el sistema base de gestión de contenido es flexible y potente. Por otra parte, el uso de este CMS en específico es una de las exigencias planteadas por el cliente del software a implementar.

#### 1.4 Conclusiones del Capítulo.

Teniendo en cuenta las consideraciones anteriormente expuestas se puede indicar que para el desarrollo de la Aplicación Web que permitirá la Gestión y Control de Métricas, se utilizará como lenguaje de programación del lado del servidor el **PHP** apoyándose en el sistema de administración de contenido **Drupal**, y el **JavaScript** del lado del cliente para lograr una mejor interactividad con el usuario en el navegador. Como Sistema de Gestión de Bases de Datos se empleará el **MySQL**, utilizando la librería ADOdb que proporciona el lenguaje PHP para el acceso a los datos, y el **Apache** como servidor Web. Durante todo el proceso de realización del sistema se utilizara como apoyo la metodología de desarrollo **XP**.

## **CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.**

### **2.1 Introducción**

En este capítulo se realiza la descripción de la situación actual referente a la evaluación y seguimiento de los proyectos productivos mediante la utilización de métricas de calidad en la UCI. Se realiza además la propuesta de las características que presentará el sistema informático que dará respuesta a la problemática existente donde se interrelacionan los procesos involucrados en la estrategia con las funcionalidades que permitirá realizar el mismo. Por otra parte se confecciona el modelo de dominio que agrupa los conceptos fundamentales en función de una mejor captura de requisitos. Se definen los requisitos funcionales y no funcionales y se confecciona un diagrama de casos de uso, donde se identifican las relaciones de los actores con el sistema.

### **2.2 Breve exposición de la situación actual.**

Para poder lograr la calidad requerida en los productos de software en la UCI mediante la evaluación de estos a través de métricas se diseñó en la universidad una estrategia que pauta la forma en que se debe proceder para lograr este propósito, la cual aun no ha podido ser utilizada en toda su extensión en el proceso de producción de software, producto a que:

- ❖ No se posee un control estricto de la información manipulada en virtud del desarrollo de los proyectos.
- ❖ Con la utilización de las métricas en cada uno de los proyectos se crea una gran cantidad de reportes, lo que dificulta la revisión de estos resultados por parte de la Dirección Central de Calidad.
- ❖ Existe desconocimiento por parte de los Líderes de proyectos y los desarrolladores acerca de todo lo relacionado con la utilización de métricas para el control de la eficiencia y calidad de los sistemas informáticos.
- ❖ No existe un ambiente adecuado donde la comunidad de la Universidad de Ciencias Informáticas pueda obtener e intercambiar información acerca de la importancia y la utilización de métricas de calidad.

Producto a esta situación es que se requiere de un sistema automatizado que permita la correcta utilización de métricas para dar una medida de la eficiencia y la calidad con que se desarrollan los proyectos durante su ciclo de desarrollo.

## 2.3 Propuesta de sistema.

Con el objetivo de obtener la calidad requerida en los productos creados en la Universidad de las Ciencias Informáticas, se ha decidido desarrollar un sistema que fusione el trabajo del **Departamento Central de Calidad** con el de cada uno de los **líderes de proyecto**. El mismo brindará amplias facilidades para almacenar y manejar toda la información referente a las **evaluaciones sistemáticas** aplicadas a cada uno de los productos de software existente en la UCI. A continuación se expondrán los diferentes servicios que ofrecerá dicho sistema. Para el trabajo con el sistema se han definido tres roles: un **usuario** que se puede comportar como **administrador**, el cual tendrá acceso a todas las funcionalidades del sistema, un grupo de usuarios que se comportará como **líder de proyecto**, este tendrá acceso a determinadas funcionalidades, aplicadas solo a los **proyectos** y **productos** de los cuales ellos sean líderes, las cuales serán asignadas por el administrador en su debido momento y por último el resto de los usuarios que pueden acceder al sistema en cualquier momentos para obtener información brindada en este o hacer uso de otros servicios brindados en el mismo, tales como foro de discusión y encuesta. La información de todos los usuarios que interactúen con el sistema se registrará y controlará en la base de datos de este.

El sistema permitirá insertar los datos de cada proyecto, acción que realizará el administrador, y una vez hecha la inserción, tanto el administrador como el líder de proyecto podrán modificarlos y consultarlos de ser necesario. Cuando el proyecto se culmine y no sea preciso tenerlo registrado el administrador podrá eliminarlo.

El sistema deberá también gestionar todo lo referente a las **métricas**, el administrador podrá tanto insertarlas, modificarlas como eliminarlas. Cada líder de proyecto tendrá la posibilidad de hacer utilización de dichas métricas para la evaluación posterior del **software** deseado. Con el objetivo de brindar una mayor información acerca de los productos, una vez evaluado estos, tanto el líder de proyecto como el administrador tendrán la posibilidad de acceder a los **reportes** arrojados al finalizar la utilización de la métrica, el líder de proyecto tendrá derecho en todo momento de visualizar el **historial de los reportes** correspondientes a su proyecto, el administrador, podrá obtener dicho historial de cada uno de los proyectos existentes en el sistema en todo momento.

Por último y no menos importante permitirá gestionar usuarios. El administrador podrá insertar, modificar, consultar y eliminar los datos de los usuarios, así como restringir su acceso a cada funcionalidad y su autenticación.

### 2.3.1 Personas relacionadas con el sistema.

Se define como persona relacionada al sistema toda aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan en el mismo. Además de todas esas que se encuentran

involucradas en dichos procesos, pues participan en ellos pero no obtienen ningún resultado de valor.

**Tabla 1 Personas relacionadas con el sistema.**

<b>Personas relacionadas con el sistema.</b>	<b>Justificación</b>
Usuario	Cualquier persona que posea una cuenta en el dominio uci.cu que interactúe con el sistema, fluye dentro de este como un invitado, sin privilegios. Tiene la posibilidad acceder a las encuestas, ver su perfil, navegar sobre las diferentes opciones para que se vaya relacionando con el mismo, así como indagar en las diferentes bibliografías que se muestran de manera general.
Líder de proyecto	Usuario que está al frente de un proyecto y puede realizar acciones de modificación y consulta solo sobre su proyecto y métricas utilizadas por este así como acceder a los reportes relacionados con su proyecto.
Administrador	Es la persona facultada para la gestión del sistema. Es el encargado de administrar las diferentes cuentas de los usuarios autenticados en la aplicación. Además tendrá la responsabilidad de insertar, modificar y eliminar tanto proyectos como métricas, así como la categoría relacionada a estas últimas.

**2.3.2 Requerimientos funcionales del sistema.**

**R1 Autenticar Usuario.**

**R2 Administrar Usuario.**

- 2.1 Asignar permisos a usuario.
- 2.2 Buscar usuario por determinados criterios.
- 2.3 Modificar permisos del usuario.

2.4 Mostrar perfil del usuario.

**R3 Gestionar Proyecto.**

3.1 Insertar proyecto.

3.2 Modificar datos del proyecto.

3.3 Eliminar proyecto.

3.4 Buscar proyecto por determinados criterios.

**R4 Gestionar Categoría.**

4.1 Insertar los datos de la categoría.

4.2 Modificar los datos de la categoría.

4.3 Eliminar categoría.

**R5 Gestionar Métrica.**

5.1 Insertar los datos de la métrica.

5.2 Modificar los datos de la métrica.

5.3 Eliminar métrica

5.4 Buscar métricas por determinados criterios.

**R6 Utilizar Métrica.**

**R7 Obtener Reporte.**

**R8 Asignar Proyecto.**

**R9 Obtener comportamiento por métrica.**

**R10 Obtener historial de proyectos por métricas.**

**2.3.3 Requerimientos no funcionales del sistema.**

**RNF 1 Apariencia:**

1.1 Diseño sencillo, de manera que se pueda utilizar el sistema sin mucho entrenamiento.

1.2 Diseño para la resolución 800x600, aunque preparado para verse en otras resoluciones.

**RNF 2 Usabilidad:**

- 2.1 El sistema podrá ser usado por cualquier persona, que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.
- 2.2 Instalar el sistema, trae consigo una mayor rapidez de trabajo y por consiguiente un ahorro de materiales de oficina y personal.

**RNF 3 Soporte:**

- 3.1 Se requiere un servidor de bases de datos con las siguientes características:
  - 3.1.1 Soporte para grandes volúmenes de datos y velocidad de procesamiento.
  - 3.1.2 Tiempo de respuesta rápido en accesos concurrentes.
- 3.2 Versión de PHP 5.0 o superior.
- 3.3 Por parte del cliente se requiere un navegador capaz de interpretar Java Script.

**RNF 4 Seguridad:**

- 4.1 Establecer permisos para el acceso garantizando que solo acceda a la información quien tenga permiso.
- 4.2 Garantizar que las funcionalidades del sistema se muestren de acuerdo a los permisos del usuario que esté logueado.
- 4.3 Garantizar la protección ante acciones no autorizadas.
- 4.4 Verificación sobre acciones irreversibles (eliminación, modificación).

**RNF 5 Confiabilidad:**

- 5.1 Garantizar un tratamiento adecuado de excepciones y validaciones de las entradas de los usuarios.
- 5.2 Garantizar la recuperación ante fallos y errores.

**RNF 6 Portabilidad:**

- 6.1 Independencia de la plataforma.

**2.4 Conclusiones del Capítulo.**

En este capítulo se hizo una breve exposición de la situación actual en el ámbito de la calidad del software en la UCI y se expusieron las razones por las cuales se pretende desarrollar el Sistema. Se

realizó la propuesta de solución se exponen además los requisitos funcionales y no funcionales de la aplicación que se pretende desarrollar, lo cual constituye el punto de partida en la construcción del Sistema.

## CAPÍTULO 3. EXPLORACIÓN Y PLANIFICACIÓN.

### 3.1 Introducción.

En este capítulo se hace énfasis en las fases de exploración y planificación propias de la metodología de desarrollo utilizadas para la implementación del sistema que se propone. Se presentan además los artefactos generados durante el transcurso de las mismas.

### 3.2 Fase de exploración.

La metodología de desarrollo comienza su trabajo en la fase de exploración, que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología. [1]

#### 3.2.1 Historias de Usuarios.

En XP debe haber predominio de algunos factores importantes como la Comunicación y el diseño Simple, esta metodología se basa en las Historias de Usuario para representar los requerimientos del sistema. En aras de la simplicidad y de no generar documentación demasiado pesada, las historias de usuario son una descripción de las necesidades funcionales que no debe ocupar muchas líneas. La idea que sea sencillo y que comunique la necesidad se cumple. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. [3]

**Tabla 2 HU Autenticar Usuario.**

Historia de Usuario	
<b>Número:</b> 1	<b>Nombre:</b> Autenticar Usuario
<b>Usuario:</b> Usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos de Estimación:</b> 0.3	<b>Iteración Asignada:</b> 1

<p><b>Descripción:</b> Se brinda la posibilidad de que la persona que acceda al sistema introduzca sus datos (usuario y contraseña) con la finalidad de verificar y otorgarle los permisos según el rol que cumpla dentro de la aplicación. Los datos asociados a cada cuenta de usuario van a coincidir con los utilizados por este para acceder al dominio UCI. CU.</p>
<p><b>Observaciones:</b> Se hace referencia al requisito R1</p>

**Tabla 3 HU Administrar Usuario.**

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre:</b> Administrar Usuario
<b>Usuario:</b> Administrador	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos de Estimación:</b> 0.4	<b>Iteración Asignada:</b> 1
<p><b>Descripción:</b> Se realizan las acciones de asignar y modificar los permisos correspondientes a usuarios del sistema. El administrador tiene la posibilidad de ver los datos de los usuarios previamente autenticados en el sistema, que serán los correspondientes a su cuenta en el dominio UCI.CU, así como de realizar la búsqueda de usuarios teniendo en cuenta algunos criterios.</p>	
<p><b>Observaciones:</b> Se hace referencia a los requisitos R2.</p>	

**Tabla 4 HU Gestionar Proyecto.**

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre:</b> Gestionar Proyecto
<b>Usuario:</b> Administrador	
<b>Prioridad en negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Medio

<b>Puntos de Estimación:</b> 1	<b>Iteración Asignada:</b> 1
<p><b>Descripción:</b> Se realizan las acciones de crear, modificar y eliminar proyectos productivos. El administrador tiene la posibilidad de insertar todos los datos relacionados al proyecto, así como acceder a estos cada vez que desee con el objetivo de modificarlos en caso que sea necesario o eliminar estos del sistema después de culminado el trabajo de dicho proyecto.</p>	
<p><b>Observaciones:</b> Se hace referencia a los requisitos R3.</p>	

**Tabla 5 HU Gestionar Categoría.**

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre:</b> Gestionar Categoría
<b>Usuario:</b> Administrador	
<b>Prioridad en negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos de Estimación:</b> 1	<b>Iteración Asignada:</b> 1
<p><b>Descripción:</b> Se realizan las acciones de crear, modificar y eliminar las Categorías asociadas a las métricas. El administrador tiene la posibilidad de insertar todos los datos relacionados a las categorías, así como acceder a estos cada vez que desee con el objetivo de modificarlos en caso que sea necesario o eliminar estos del sistema en el momento que el administrador crea que dicha categoría no será utilizada nuevamente.</p>	
<p><b>Observaciones:</b> Se hace referencia a los requisitos R4.</p>	

**Tabla 6 HU Gestionar Métrica.**

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre:</b> Gestionar Métrica
<b>Usuario:</b> Administrador	

<b>Prioridad en negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Alto
<b>Puntos de Estimación:</b> 1	<b>Iteración Asignada:</b> 2
<p><b>Descripción:</b> Se realizan las acciones de crear, modificar y eliminar las métricas deseadas. El administrador tiene la posibilidad de insertar todos los datos relacionados a dichas métricas, así como realizar una búsqueda de estas siguiendo una serie de parámetros con el objetivo de modificar sus datos en caso que sea necesario o eliminar estas completamente del sistema en el momento que el administrador crea que ya no serán utilizadas.</p>	
<p><b>Observaciones:</b> Se hace referencia a los requisitos R5.</p>	

**Tabla 7 HU Utilizar Métrica.**

<b>Historia de Usuario</b>	
<b>Número:</b> 6	<b>Nombre:</b> Utilizar Métrica
<b>Usuario:</b> Líder de proyecto.	
<b>Prioridad en negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Alto
<b>Puntos de Estimación:</b> 2	<b>Iteración Asignada:</b> 2
<p><b>Descripción:</b> El líder de proyecto tendrá la posibilidad de utilizar la métrica que necesite, con el objetivo de llevar un chequeo sistemático del desarrollo y rendimiento de su proyecto.</p>	
<p><b>Observaciones:</b> Se hace referencia a los requisitos R6.</p>	

**Tabla 8 HU Obtener Reporte.**

<b>Historia de Usuario</b>	
<b>Número:</b> 7	<b>Nombre:</b> Obtener Reporte
<b>Usuario:</b> Líder de Proyecto	

<b>Prioridad en negocio:</b> Medio	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos de Estimación:</b> 1	<b>Iteración Asignada:</b> 2
<b>Descripción:</b> El líder de proyecto tendrá acceso al resultado arrojado por la utilización de una métrica determinada, de esta misma forma podrá observar los resultados anteriores de esta, también podrá tener acceso a una gráfica que representa todos estos resultados, con el objetivo de obtener una comparación de estos.	
<b>Observaciones:</b> Se hace referencia a los requisitos R7.	

**Tabla 9 HU Asignar Proyecto.**

Historia de Usuario	
<b>Número:</b> 8	<b>Nombre:</b> Asignar Proyecto.
<b>Usuario:</b> Administrador	
<b>Prioridad en negocio:</b> Medio	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos de Estimación:</b> 1	<b>Iteración Asignada:</b> 3
<b>Descripción:</b> El Administrador podrá acceder a una lista de los líderes de proyecto que no estén asociados a ningún proyecto productivo, y de esta forma podrá vincularlos, así dicho usuario podrá hacer uso de las métricas, aplicándolas al proyecto al que pertenece.	
<b>Observaciones:</b> Se hace referencia a los requisitos R8.	

**Tabla 10 HU Obtener Comportamiento por Métrica.**

Historia de Usuario	
<b>Número:</b> 9	<b>Nombre:</b> Obtener comportamiento por métrica.
<b>Usuario:</b> Líder de Proyecto	

<b>Prioridad en negocio:</b> Medio	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos de Estimación:</b> 1	<b>Iteración Asignada:</b> 3
<b>Descripción:</b> El Líder de proyecto podrá acceder a una lista donde estarán todas las métricas que han sido utilizadas para evaluar el proyecto al que este pertenece, dándole la posibilidad de escoger una y de esta forma observar el comportamiento de dicho proyecto según la métrica escogida.	
<b>Observaciones:</b> Se hace referencia a los requisitos R9.	

**Tabla 11 HU Obtener Historial de Proyecto por Métrica.**

Historia de Usuario	
<b>Número:</b> 10	<b>Nombre:</b> Obtener Historial de Proyecto por Métrica.
<b>Usuario:</b> Administrador	
<b>Prioridad en negocio:</b> Medio	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos de Estimación:</b> 1	<b>Iteración Asignada:</b> 3
<b>Descripción:</b> El administrador podrá acceder a una lista donde estarán todos los proyectos productivos existentes en la universidad, donde tendrá la posibilidad de escoger el que desee con el objetivo de acceder a todas las métricas utilizadas por este y así posteriormente ver todos los resultados arrojados por la evaluación de dicho proyecto con la métrica seleccionada.	
<b>Observaciones:</b> Se hace referencia a los requisitos R10.	

### 3.3 Fase de planificación.

En esta fase el cliente establece la prioridad de cada historia de usuario, los programadores realizan una estimación del esfuerzo necesario para la implementación de cada una de ellas. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración,

basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. [6]

**3.3.1 Estimación de esfuerzo por historias de usuario.**

Se ha realizado una estimación para cada una de las historias de usuario previamente identificadas, con el objetivo de llevar a realizar un buen desarrollo del sistema propuesto, los resultados de dicha estimación se muestran a continuación:

**Tabla 12 Estimación del esfuerzo por Historias de Usuario.**

Historia de Usuario	Puntos de Estimación
Autenticar Usuario	0.3
Administrar Usuario	0.4
Gestionar Proyecto	1
Gestionar Categoría	1
Gestionar Métrica	1
Utilizar Métrica	2
Obtener Reporte	1
Asignar Proyecto	1
Obtener Comportamiento por Métrica	1
Obtener Historial de Proyecto por Métrica	1

**3.3.2 Plan de Iteraciones.**

Todo proyecto que siga la metodología XP se ha de dividir en iteraciones de aproximadamente 3 semanas de duración, por lo tanto luego de la previa identificación y descripción de las historias de

usuario y realizar la estimación de esfuerzo de cada una de ellas se procede a la planificación de la etapa de implementación del sistema, donde se definen cuales historias de usuario serán implementadas en cada iteración del proyecto así como las posibles fechas de culminación.

Se decidió realizar un total de tres iteraciones que abarcaran la implementación de la totalidad de historias de usuario, cada una de estas se define a continuación:

### **Iteración 1**

El objetivo de esta iteración es la implementación de aquellas historias de usuario que son consideradas las de mayor prioridad para el sistema, a medida que se lleva a cabo el desarrollo de la misma se consolidará la base de la arquitectura del sistema. Como resultado de esta iteración se contará con las funcionalidades descritas en las Historias de usuario 1, 2, 3, y 4 las cuales hacen referencia a la autenticación de usuarios en el sistema, la administración de usuarios del mismo, así como la gestión de proyectos y categorías de este. Además se tendrá la primera versión de prueba, la cual será mostrada al cliente con el objetivo de obtener una retroalimentación para el grupo de trabajo.

### **Iteración 2**

Durante esta iteración se implementan las restantes funcionalidades con prioridad alta, al terminar esta se habrá implementado lo relacionado con las Historias de usuario 5 y 6, en las cuales se relacionan con la gestión y utilización de métricas de calidad. La versión de prueba referente a esta iteración junto a las implementaciones anteriores, serán mostradas al cliente con el objetivo de realizar cambios en base a la opinión del mismo.

### **Iteración 3**

En esta iteración serán implementadas el resto de las funcionalidades, estas hacen alusión a todo lo relacionado con los reportes asociados a los proyectos y las métricas utilizadas para la evaluación de estos. Estas funciones están descritas en las Historias de usuario 7, 8, 9,10 y 11 respectivamente. Como resultado de esta iteración se tendrá la versión 1.0 del producto final, adicionando lo concerniente a la interfaz Web. A partir de este momento el sistema será puesto a prueba por un período de tiempo para evaluar el desempeño del mismo.

#### **3.3.3 Plan de duración de las iteraciones.**

Para la implementación del sistema se cuenta con un único equipo de desarrollo, debido a esto se ha creado un plan de desarrollo para este, el cual tiene la finalidad de mostrar la duración de cada iteración basándose en el orden y el tiempo estimado de cada una de las historias de usuario

**Tabla 13 Plan de duración de las iteraciones.**

Iteraciones	Orden de las Historias de usuario a implementar	Duración total de las iteraciones
Iteración 1	1- Autenticar Usuario. 2- Administrar Usuario. 3- Gestionar Proyecto. 4- Gestionar Categoría.	3 semanas
Iteración 2	1- Gestionar Métrica. 2- Utilizar Métrica. 3- Obtener Reporte	4 semanas
Iteración 3	1- Asignar Proyecto 2- Obtener Comportamiento por Métrica. 3- Obtener Historial de Proyecto por Métrica.	3 semanas

**3.3.4 Plan de entregas.**

Para la fase de implementación se ha elaborado un plan de entregas, con el objetivo de facilitar la elaboración de estas todas las funcionalidades con puntos en común fueron agrupadas en módulos, quedando estos conformados de la siguiente forma:

**Tabla 14 Módulos y HU abarcadas**

Módulos	Historias de usuario que abarca
user	1- Autenticar Usuario 2- Administrar Usuario
proyecto	1- Gestionar Proyecto
categoría	1- Gestionar categoría

métrica	1- Gestionar Métrica
resultado	1- Utilizar Métrica 2- Obtener Reporte
administrar	1- Asignar Proyecto
reporte	1- Obtener Historial de Proyecto por Métrica
reporte_lider	2- Obtener Comportamiento por Métrica

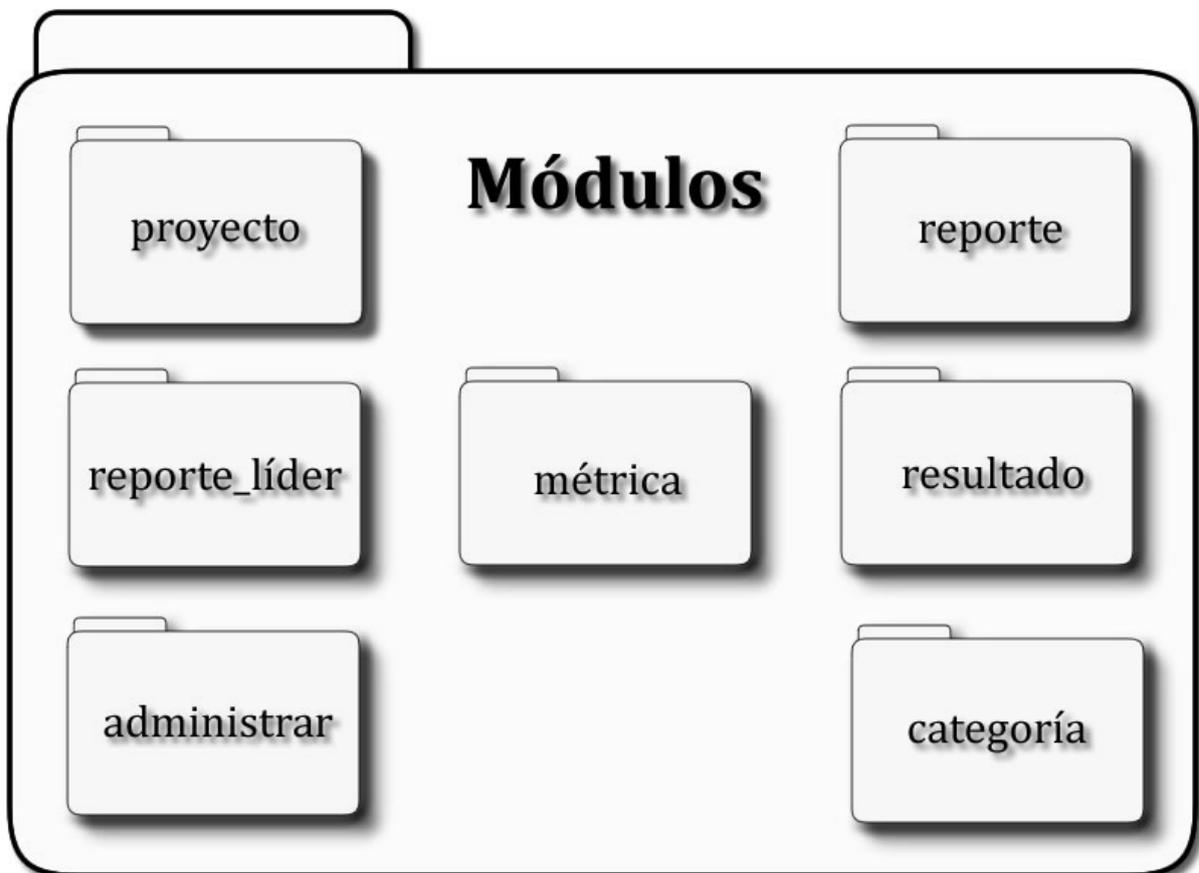
Como producto del plan de entregas se harán releases al sistema en las fechas que se indican a continuación:

**Tabla 15 Plan de duración entrega.**

Módulos	Final 1ra iteración 1ra semana de Abril	Final 2da iteración 1ra semana de Mayo	Final 3ra iteración 4ta semana de Mayo
user	1.0		
proyecto	1.0		
categoría	1.0		
métrica		1.0	
resultado		3.0	
administrar			1.0
reporte			1.0
reporte_lider			1.0

### 3.4 Módulos Implementados.

Debido a la naturaleza de los Sistemas de Gestión de Contenidos y a su estructura modular, la cual nos permite extender y moldear a gusto del administrador distintas funciones por medio de módulos de fácil instalación, en esta sección se presentará un modelo con todos los módulos implementados los cuales ofrecen la oportunidad de dar cumplimiento a todos los requerimientos que el sistema necesita.



Con el objetivo de hacer entendible las funcionalidades encapsuladas en estos módulos, se define una tarjeta CRC por cada uno, con la finalidad de obtener un diseño simple y no incurrir en la implementación de características que no son necesarias.

**Tabla 16 Tarjeta CRC Módulo proyecto**

Módulo proyecto	
Funcionalidades	Colaboraciones (Módulos)

Adicionar proyecto.	Node
Eliminar proyecto.	Block
Modificar datos del proyecto.	User
Buscar proyecto por determinados criterios.	Proyecto

**Tabla 17 Tarjeta CRC Módulo reporte**

Módulo reporte	
Funcionalidades	Colaboraciones (Módulos)
Obtener reporte	Node Categoría Métrica User Block Reporte

**Tabla 18 Tarjeta CRC Módulo resultado.**

Módulo resultado	
Funcionalidades	Colaboraciones (Módulos)
Utilizar Métrica Obtener Reporte	Node categoría métrica User Block resultado administrar proyecto

**Tabla 19 Tarjeta CRC Módulo categoría.**

Módulo categoría	
Funcionalidades	Colaboraciones (Módulos)

Adicionar categoría.	Node
Eliminar categoría.	Block
Modificar datos del una categoría.	User
	Categoría

**Tabla 20 Tarjeta CRC Módulo administrar.**

Módulo administrar	
Funcionalidades	Colaboraciones (Módulos)
Asignar proyecto.	Node Block User Proyecto

**Tabla 21 Tarjeta CRC Módulo reporte\_lider.**

Módulo reporte_lider	
Funcionalidades	Colaboraciones (Módulos)
Obtener Comportamiento por Métrica.	Node Block User métrica resultado administrar

**Tabla 22 Tarjeta CRC Módulo métrica.**

Módulo métrica	
Funcionalidades	Colaboraciones (Módulos)
Adicionar métrica.	Node
Eliminar métrica.	Block
Modificar datos de una métrica	User
Buscar métricas por determinados criterios.	métrica

	categoria
--	-----------

### 3.5 Conclusiones del Capítulo.

Durante el desarrollo de este capítulo se hizo énfasis en todo lo relacionado con las fases de exploración y planificación del proyecto, describiéndose además cada uno de los artefactos generados durante el transcurso de las mismas.

## **CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA.**

### **4.1 Introducción.**

El desarrollo de un software con la metodología XP se debe llevar a cabo de forma iterativa obteniendo al finalizar cada iteración un prototipo funcional para que el cliente pueda probarlo y dar su opinión acerca del mismo, con el objetivo de incrementar la visión de los desarrolladores con la opinión de este. En este capítulo se detallan las tres iteraciones llevadas a cabo durante el desarrollo del sistema, exponiendo las tareas generadas al analizar cada una de las historias de usuario, así como las pruebas de aceptación efectuadas sobre el sistema.

### **4.2 Fase de Implementación.**

Durante cada una de las iteraciones se lleva a cabo la implementación de las historias de usuario seleccionadas para ser realizadas en cada una de ellas. Cada historia de usuario se divide en tareas de desarrollo, que serán asignadas posteriormente a un grupo de desarrollo que será responsable de su implementación. Estas tareas son para el uso estricto de los programadores, pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente.

A continuación se detallan cada una de las tres iteraciones, teniendo en cuenta la planificación realizada:

#### **4.2.1 Iteración 1**

En esta iteración se implementan las historias de usuario de mayor prioridad, algunas necesarias para la asignación de permisos y otras imprescindibles para la posterior implementación de otras, con la culminación de ésta se desea que cualquier usuario pueda acceder a la aplicación haciendo uso de su cuenta en el dominio UCI.CU.

**Tabla 23 Módulos abordados en la primera iteración**

Módulo	Historias de usuario	Tiempo de Implementación (semana)	
		Estimación	Real
user	Autenticar Usuario	0.3	0.3
	Administrar Usuario	0.4	0.4
proyecto	Gestionar Proyecto	1	1
categoría	Gestionar categoría	1	1

A continuación se muestran las tareas relacionadas con las historias de usuario relacionadas con cada uno de los módulos implementados en esta iteración:

**Módulo user**

**Tabla 24 Tarea # 1 del módulo user.**

<b>Tarea</b>	
<b>Número de tarea:</b> 1	<b>Número de HU:</b> 1
<b>Nombre de la tarea:</b> Configuración de la interfaz para la autenticación de usuarios	
<b>Tipo de tarea:</b> Configuración	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 17 de Marzo del 2008	<b>Fecha fin:</b> 18 de Marzo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se configura la interfaz de inicio en la cual se podrán autenticar los usuarios de la aplicación.	

**Tabla 25 Tarea # 2 del módulo user.**

<b>Tarea</b>
--------------

<b>Número de tarea:</b> 2	<b>Número de HU:</b> 1
<b>Nombre de la tarea:</b> Captura de los datos de autenticación	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 18 de Marzo del 2008	<b>Fecha fin:</b> 19 de Marzo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Los datos de los usuarios se capturan para su posterior comparación.	

**Tabla 26 Tarea # 3 del módulo user.**

<b>Tarea</b>	
<b>Número de tarea:</b> 3	<b>Número de HU:</b> 1
<b>Nombre de la tarea:</b> Verificación de permisos.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.05
<b>Fecha inicio:</b> 19 de Marzo del 2008	<b>Fecha fin:</b> 20 de Marzo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se comprueban los permisos correspondientes al usuario autenticado.	

**Tabla 27 Tarea # 4 del módulo user.**

<b>Tarea</b>	
<b>Número de tarea:</b> 4	<b>Número de HU:</b> 1
<b>Nombre de la tarea:</b> Acceso a la aplicación con los permisos correspondientes.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.05

<b>Fecha inicio:</b> 19 de Marzo del 2008	<b>Fecha fin:</b> 20 de Marzo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se permite el acceso al usuario con los permisos correspondientes.	

Tabla 28 Tarea # 5 del módulo user.

Tarea	
<b>Número de tarea:</b> 5	<b>Número de HU:</b> 2
<b>Nombre de la tarea:</b> Configuración de la interfaz para administrar usuario.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 20 de Marzo del 2008	<b>Fecha fin:</b> 21 de Marzo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se configura la interfaz para la asignación de permisos de un usuario determinado que hará uso de la aplicación.	

Tabla 29 Tarea # 6 del módulo user.

Tarea	
<b>Número de tarea:</b> 6	<b>Número de HU:</b> 2
<b>Nombre de la tarea:</b> Buscar Usuario.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 21 de Marzo del 2008	<b>Fecha fin:</b> 22 de Marzo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	

**Descripción:** Se busca el usuario deseado para su posterior asignación de permisos.

**Tabla 30 Tarea # 7 del módulo user.**

Tarea	
<b>Número de tarea:</b> 7	<b>Número de HU:</b> 2
<b>Nombre de la tarea:</b> Asignar responsabilidad a usuario.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 22 de Marzo del 2008	<b>Fecha fin:</b> 23 de Marzo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se le asigna la responsabilidad correspondiente al usuario seleccionado.	

**Tabla 31 Tarea # 8 del módulo user.**

Tarea	
<b>Número de tarea:</b> 8	<b>Número de HU:</b> 2
<b>Nombre de la tarea:</b> Guardar en la base de datos los cambios efectuados.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 23 de Marzo del 2008	<b>Fecha fin:</b> 24 de Marzo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se guarda en la base de datos las responsabilidades asignadas a los usuarios, con el objetivo que puedan acceder a los módulos correspondientes.	

**Tabla 32 Tarea # 1 del módulo proyecto.**

Tarea	
<b>Número de tarea:</b> 1	<b>Número de HU:</b> 3
<b>Nombre de la tarea:</b> Configuración de la interfaz para insertar proyecto.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha inicio:</b> 24 de Marzo del 2008	<b>Fecha fin:</b> 26 de Marzo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se configura la interfaz para que el usuario pueda insertar todos los datos correspondientes al proyecto.	

**Tabla 33 Tarea # 2 del módulo proyecto.**

Tarea	
<b>Número de tarea:</b> 2	<b>Número de HU:</b> 3
<b>Nombre de la tarea:</b> Verificar la validez de los datos insertados por el usuario.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 26 de Marzo del 2008	<b>Fecha fin:</b> 27 de Marzo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se verifica si el usuario ha insertado los datos correctamente y si el proyecto deseado no existe en la base de datos.	

**Tabla 34 Tarea # 3 del módulo proyecto.**

Tarea	
<b>Número de tarea:</b> 3	<b>Número de HU:</b> 3
<b>Nombre de la tarea:</b> Guardar en la base de datos el proyecto deseado.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha inicio:</b> 27 de Marzo del 2008	<b>Fecha fin:</b> 29 de Marzo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se guarda en la base de datos las características correspondientes al proyecto.	

**Tabla 35 Tarea # 4 del módulo proyecto.**

Tarea	
<b>Número de tarea:</b> 4	<b>Número de HU:</b> 3
<b>Nombre de la tarea:</b> Eliminar proyecto.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 29 de Marzo del 2008	<b>Fecha fin:</b> 30 de Marzo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se elimina de la base de datos un proyecto deseado.	

**Tabla 36 Tarea # 5 del módulo proyecto.**

Tarea
-------

<b>Número de tarea:</b> 5	<b>Número de HU:</b> 3
<b>Nombre de la tarea:</b> Modificar datos de proyecto.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 30 de Marzo del 2008	<b>Fecha fin:</b> 31 de Marzo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se modifican en la base de datos las características del proyecto deseado.	

**Tabla 37 Tarea # 1 del módulo categoría.**

<b>Tarea</b>	
<b>Número de tarea:</b> 1	<b>Número de HU:</b> 4
<b>Nombre de la tarea:</b> Configuración de la interfaz para insertar categoría.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha inicio:</b> 31 de Marzo del 2008	<b>Fecha fin:</b> 2 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se configura la interfaz para que el usuario pueda insertar todos los datos correspondientes a una categoría.	

**Tabla 38 Tarea # 2 del módulo categoría.**

<b>Tarea</b>	
<b>Número de tarea:</b> 2	<b>Número de HU:</b> 4
<b>Nombre de la tarea:</b> Verificar la validez de los datos insertados por el usuario.	

<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 2 de Abril del 2008	<b>Fecha fin:</b> 3 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se verifica si el usuario ha insertado los datos correctamente y si la categoría deseada no existe en la base de datos.	

**Tabla 39 Tarea # 3 del módulo categoría.**

Tarea	
<b>Número de tarea:</b> 3	<b>Número de HU:</b> 4
<b>Nombre de la tarea:</b> Guardar en la base de datos la categoría deseada.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha inicio:</b> 3 de Abril del 2008	<b>Fecha fin:</b> 5 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se guarda en la base de datos las características correspondientes a la categoría.	

**Tabla 40 Tarea # 4 del módulo categoría.**

Tarea	
<b>Número de tarea:</b> 4	<b>Número de HU:</b> 4
<b>Nombre de la tarea:</b> Eliminar categoría.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 5 de Abril del 2008	<b>Fecha fin:</b> 6 de Abril del 2008

<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López
<b>Descripción:</b> Se elimina de la base de datos una categoría deseada.

**Tabla 41 Tarea # 5 del módulo categoría.**

Tarea	
<b>Número de tarea:</b> 5	<b>Número de HU:</b> 4
<b>Nombre de la tarea:</b> Modificar datos de una categoría.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 6 de Abril del 2008	<b>Fecha fin:</b> 7 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se modifican en la base de datos las características de la categoría deseada.	

#### 4.2.2 Iteración 2

En esta iteración se desarrollan algunas historias de usuario de alta prioridad en el desarrollo del software, con el termino de ésta se desea contar con la culminación de las principales funcionalidades, de forma tal que tanto los administradores como los líderes de proyecto puedan hacer todo lo necesario para evaluar un proyecto de acuerdo a las métricas que deseen utilizar.

**Tabla 42 Módulos abordados en la segunda iteración.**

Módulo	Historias de usuario	Tiempo de Implementación	
		Estimación	Real
métrica	Gestionar Métrica	1	1
resultado	Utilizar Métrica	2	2
	Obtener Reporte	1	1

**Tabla 43 Tarea # 1 del módulo métrica.**

Tarea	
<b>Número de tarea:</b> 1	<b>Número de HU:</b> 5
<b>Nombre de la tarea:</b> Configuración de la interfaz para insertar métrica.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha inicio:</b> 7 de Abril del 2008	<b>Fecha fin:</b> 9 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se configura la interfaz para que el usuario pueda insertar todos los datos correspondientes a la métrica.	

**Tabla 44 Tarea # 2 del módulo métrica**

Tarea	
<b>Número de tarea:</b> 2	<b>Número de HU:</b> 5
<b>Nombre de la tarea:</b> Verificar la validez de los datos insertados por el usuario.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha inicio:</b> 9 de Abril del 2008	<b>Fecha fin:</b> 11 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se verifica si el usuario ha insertado los datos correctamente y si la métrica deseada no existe en la base de datos.	

**Tabla 45 Tarea # 3 del módulo métrica.**

Tarea	
<b>Número de tarea:</b> 3	<b>Número de HU:</b> 5
<b>Nombre de la tarea:</b> Guardar en la base de datos el proyecto deseado.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 11 de Abril del 2008	<b>Fecha fin:</b> 12 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se guarda en la base de datos los datos correspondientes a la métrica.	

**Tabla 46 Tarea # 4 del módulo métrica.**

Tarea	
<b>Número de tarea:</b> 4	<b>Número de HU:</b> 5
<b>Nombre de la tarea:</b> Eliminar métrica.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 12 de Abril del 2008	<b>Fecha fin:</b> 13 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se elimina de la base de datos la métrica deseada.	

Tabla 47 Tarea # 5 del módulo métrica.

Tarea	
<b>Número de tarea:</b> 5	<b>Número de HU:</b> 5
<b>Nombre de la tarea:</b> Modificar datos de una métrica.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 13 de Abril del 2008	<b>Fecha fin:</b> 14 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se modifican en la base de datos las características de una métrica deseada.	

Tabla 48 Tarea # 1 del módulo resultado.

Tarea	
<b>Número de tarea:</b> 1	<b>Número de HU:</b> 6
<b>Nombre de la tarea:</b> Configuración de la interfaz para utilizar una métrica.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.3
<b>Fecha inicio:</b> 14 de Abril del 2008	<b>Fecha fin:</b> 17 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se configura la interfaz para que el usuario escoger tanto la métrica que desea utilizar como para que introduzca todos los valores de la fórmula correspondiente.	

**Tabla 49 Tarea # 2 del módulo resultado.**

Tarea	
<b>Número de tarea:</b> 2	<b>Número de HU:</b> 6
<b>Nombre de la tarea:</b> Verificar la validez de los datos insertados por el usuario.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.3
<b>Fecha inicio:</b> 17 de Abril del 2008	<b>Fecha fin:</b> 20 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se verifica si el usuario ha insertado todos los datos necesarios y lo ha hecho correctamente.	

**Tabla 50 Tarea # 3 del módulo resultado.**

Tarea	
<b>Número de tarea:</b> 3	<b>Número de HU:</b> 6
<b>Nombre de la tarea:</b> Calcular resultado.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 20 de Abril del 2008	<b>Fecha fin:</b> 21 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se calcula el resultado, según la métrica escogida y los valores insertados por el usuario.	

Tabla 51 Tarea # 3 del módulo resultado.

Tarea	
<b>Número de tarea:</b> 4	<b>Número de HU:</b> 6
<b>Nombre de la tarea:</b> Guardar en la base el resultado arrojado.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.1
<b>Fecha inicio:</b> 21 de Abril del 2008	<b>Fecha fin:</b> 22 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se guarda en la base de datos el resultado obtenido como consecuencia de la utilización de una métrica para evaluar un proyecto productivo.	

Tabla 52 Tarea # 1 del módulo resultado.

Tarea	
<b>Número de tarea:</b> 1	<b>Número de HU:</b> 7
<b>Nombre de la tarea:</b> Configuración de la interfaz para observar reporte.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha inicio:</b> 22 de Abril del 2008	<b>Fecha fin:</b> 24 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se configura la interfaz para que el usuario pueda ver los últimos resultados obtenidos mediante el uso de la métrica que acaba de utilizar.	

Tabla 53 Tarea # 2 del módulo resultado.

Tarea	
<b>Número de tarea:</b> 2	<b>Número de HU:</b> 7
<b>Nombre de la tarea:</b> Mostrar los últimos resultados de la métrica utilizada.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha inicio:</b> 24 de Abril del 2008	<b>Fecha fin:</b> 26 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se muestran los últimos resultados de la aplicación de la métrica a un proyecto determinado, así como la fecha en que se obtuvieron.	

Tabla 54 Tarea # 3 del módulo resultado.

Tarea	
<b>Número de tarea:</b> 3	<b>Número de HU:</b> 7
<b>Nombre de la tarea:</b> Mostrar los últimos resultados de la métrica utilizada mediante una gráfica.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.3
<b>Fecha inicio:</b> 26 de Abril del 2008	<b>Fecha fin:</b> 29 de Abril del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se le da la posibilidad al usuario de ver los últimos resultados de la aplicación de la métrica a un proyecto determinado, así como la fecha en que se obtuvieron mediante una gráfica, con el objetivo de que pueda comparar estos de una forma más fácil.	

### 4.2.3 Iteración 3

En esta iteración se desarrollan las historias de usuario de menor complejidad, con la culminación de esta se desea que tanto el administrador puedan acceder a los reportes a los cuales cada uno posee acceso, el principal objetivo de esta iteración es dar culminación al desarrollo de la aplicación.

**Tabla 55 Módulos abordados en la tercera iteración.**

Módulo	Historias de usuario	Tiempo de Implementación (semana)	
		Estimación	Real
administrar	Asignar Proyecto	1.0	1.0
reporte	Obtener Historial de Proyecto por Métrica	1.0	1.0
reporte_lider	Obtener Comportamiento por Métrica	1.0	1.0

A continuación se muestran las tareas relacionadas con las historias de usuario relacionadas con cada uno de los módulos implementados en esta iteración:

**Tabla 56 Tarea # 1 del módulo administrar.**

Tarea	
<b>Número de tarea:</b> 1	<b>Número de HU:</b> 8
<b>Nombre de la tarea:</b> Configuración de la interfaz para asignar un proyecto a un líder de proyecto.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha inicio:</b> 29 de Abril del 2008	<b>Fecha fin:</b> 3 de Mayo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se configura la interfaz para que el usuario pueda ver los líderes de proyecto que no poseen ninguno asociado y pueda escoger el que desee relacionar con algún proyecto en específico.	

Tabla 57 Tarea # 2 del módulo administrar.

Tarea	
<b>Número de tarea:</b> 2	<b>Número de HU:</b> 8
<b>Nombre de la tarea:</b> Guardar en la base de datos la relación existente entre proyecto y líder.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.3
<b>Fecha inicio:</b> 3 de Mayo del 2008	<b>Fecha fin:</b> 6 de Mayo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se guarda en la base de datos la relación existente entre el proyecto y su líder asociado, de esta forma éste último puede hacer uso de las métricas existentes.	

Tabla 58 Tarea # 1 del módulo reporte.

Tarea	
<b>Número de tarea:</b> 1	<b>Número de HU:</b> 10
<b>Nombre de la tarea:</b> Configuración de la interfaz para obtener el historial de proyecto por métrica.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha inicio:</b> 6 de Mayo del 2008	<b>Fecha fin:</b> 8 de Mayo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se configura la interfaz para que el usuario pueda escoger alguna de las métricas utilizadas por un proyecto y de esta forma ver todos los resultados asociados a ésta.	

Tabla 59 Tarea # 2 del módulo reporte.

Tarea	
<b>Número de tarea:</b> 2	<b>Número de HU:</b> 10
<b>Nombre de la tarea:</b> Mostrar los últimos resultados del proyecto y la métrica escogida.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha inicio:</b> 8 de Mayo del 2008	<b>Fecha fin:</b> 10 de Mayo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se muestran los últimos resultados de la aplicación de la métrica, así como la fecha en que se obtuvieron.	

Tabla 60 Tarea # 3 del módulo reporte.

Tarea	
<b>Número de tarea:</b> 3	<b>Número de HU:</b> 10
<b>Nombre de la tarea:</b> Mostrar los últimos resultados de la métrica escogida mediante una gráfica.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.3
<b>Fecha inicio:</b> 10 de Mayo del 2008	<b>Fecha fin:</b> 13 de Mayo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se le da la posibilidad al usuario de ver los últimos resultados de la aplicación de la métrica escogida así como la fecha en que se obtuvieron mediante una gráfica, con el objetivo de que pueda comparar estos de una forma más fácil.	

**Tabla 61 Tarea # 1 del módulo reporte\_lider.**

Tarea	
<b>Número de tarea:</b> 1	<b>Número de HU:</b> 9
<b>Nombre de la tarea:</b> Configuración de la interfaz para obtener el comportamiento de una métrica.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha inicio:</b> 13 de Mayo del 2008	<b>Fecha fin:</b> 15 de Mayo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se configura la interfaz para que el usuario pueda escoger alguna de las métricas utilizadas por el proyecto al que este se encuentra asociado.	

**Tabla 62 Tarea # 2 del módulo reporte\_lider.**

Tarea	
<b>Número de tarea:</b> 2	<b>Número de HU:</b> 9
<b>Nombre de la tarea:</b> Mostrar los últimos resultados de la métrica escogida.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha inicio:</b> 15 de Mayo del 2008	<b>Fecha fin:</b> 17 de Mayo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se muestran los últimos resultados de la aplicación de la métrica al proyecto asociado al usuario, así como la fecha en que se obtuvieron.	

Tabla 63 Tarea # 3 del módulo reporte\_lider.

Tarea	
<b>Número de tarea:</b> 3	<b>Número de HU:</b> 9
<b>Nombre de la tarea:</b> Mostrar los últimos resultados de la métrica escogida mediante una gráfica.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.3
<b>Fecha inicio:</b> 17 de Mayo del 2008	<b>Fecha fin:</b> 20 de Mayo del 2008
<b>Programador responsable:</b> Andry Leal Ortiz -- Leyvis Luis Valdes López	
<b>Descripción:</b> Se muestran mediante una gráfica los últimos resultados de la aplicación de la métrica al proyecto asociado al usuario, así como la fecha en que se obtuvieron.	

### 4.3 Pruebas

Anteriormente las pruebas de software se consideraban sólo una actividad que realizaba el desarrollador para encontrar fallas en sus productos; con el paso de los años se ha determinado la importancia que tienen para garantizar el tiempo, el costo y la calidad del producto, de tal forma que actualmente son un proceso cuyo propósito principal es evaluar en todo momento la generación del software respecto de los requerimientos establecidos al inicio. Cada vez que se desea implementar una funcionalidad se debe redactar una prueba sencilla, de forma tal que se pueda probar que esta luego de su supuesta culminación. Probando constantemente el software se desarrolla un producto con mayor calidad dando a los desarrolladores una mayor seguridad en el trabajo que llevan a cabo, sin margen a pensar en el que ya hayan terminado. En XP hay dos tipos de pruebas; las unitarias, desarrolladas por los programadores verificando su código de forma automática y las de aceptación las cuales son evaluadas luego de culminar una iteración verificando así si se cumplió la funcionalidad requerida por el cliente. Con estas normas se obtiene un código simple y funcional de manera bastante rápida. Por esto es importante pasar las pruebas al 100%. [10]

### 4.4 Pruebas de aceptación

Las llamadas pruebas de aceptación en la metodología XP no son más que pruebas de caja negra. Las historias de usuario definidas anteriormente en cada iteración se traducen en pruebas de

aceptación detallando desde la perspectiva del cliente que cada historia de usuario ha sido correctamente implementada, garantizando que los requisitos han sido cumplidos de forma correcta. Cada historia de usuario puede tener la cantidad de pruebas de aceptación que se requieran.

**Tabla 64 Prueba de aceptación para la historia de usuario Autenticar Usuario.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU1_P1	<b>Historia de Usuario:</b> 1 – Autenticar Usuario
<b>Nombre:</b> Autenticar un usuario al sistema	
<b>Descripción:</b> Probar que el usuario acceda correctamente al sistema con la misma cuenta que posee en el dominio UCI.CU	
<b>Condiciones de Ejecución:</b> Se insertan datos correspondientes a un usuario del dominio UCI.CU	
<b>Entrada / Pasos de ejecución:</b> Se intenta insertar un usuario con datos válidos.	
<b>Resultado Esperado:</b> El usuario es aceptado correctamente y guardado en la base de datos del sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 65 Prueba de aceptación para la historia de usuario Administrar Usuario.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU2_P1	<b>Historia de Usuario:</b> 2 – Administrar Usuario
<b>Nombre:</b> Administrar Usuario	
<b>Descripción:</b> Darle permisos de administrador o líder de proyecto a un usuario previamente registrado.	
<b>Condiciones de Ejecución:</b> El usuario debe haberse registrado anteriormente.	
<b>Entrada / Pasos de ejecución:</b> Se intenta darle los permisos correspondientes.	
<b>Resultado Esperado:</b> El usuario puede acceder a todos los servicios de acuerdo con el rol asignado.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 66 Prueba de aceptación para la historia de usuario Gestionar Proyecto.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU3_P1	<b>Historia de Usuario:</b> 3 – Gestionar Proyecto
<b>Nombre:</b> Insertar un proyecto al sistema	
<b>Descripción:</b> Probar que se inserte un proyecto al sistema	
<b>Condiciones de Ejecución:</b> Todos los datos son insertados correctamente y que la sesión esté iniciada por el administrador.	
<b>Entrada / Pasos de ejecución:</b> Se intenta guardar el proyecto en la base de datos.	
<b>Resultado Esperado:</b> El proyecto es insertado correctamente y guardado en la base de datos del sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 67 Prueba de aceptación para la historia de usuario Gestionar Proyecto.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU3_P2	<b>Historia de Usuario:</b> 3 – Gestionar Proyecto
<b>Nombre:</b> Modificar un proyecto existente en el sistema.	
<b>Descripción:</b> Se intenta modificar un proyecto existente en el sistema.	
<b>Condiciones de Ejecución:</b> Todos los datos son insertados correctamente y que la sesión esté iniciada por el administrador.	
<b>Entrada / Pasos de ejecución:</b> Se intenta guardar los nuevos datos del proyecto en la base de datos.	
<b>Resultado Esperado:</b> Los nuevos datos son guardados correctamente en la base de datos del sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 68 Prueba de aceptación para la historia de usuario Gestionar Proyecto.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU3_P3	<b>Historia de Usuario:</b> 3 – Gestionar Proyecto
<b>Nombre:</b> Eliminar un proyecto del sistema	
<b>Descripción:</b> Se desea eliminar un proyecto del sistema	
<b>Condiciones de Ejecución:</b> La sesión debe ser iniciada por el administrador.	
<b>Entrada / Pasos de ejecución:</b> Se intenta eliminar el proyecto de la base de datos.	
<b>Resultado Esperado:</b> El proyecto es eliminado correctamente de la base de datos del sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 69 Prueba de aceptación para la historia de usuario Gestionar Categoría.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU4_P1	<b>Historia de Usuario:</b> 4 – Gestionar Categoría
<b>Nombre:</b> Insertar una categoría al sistema	
<b>Descripción:</b> Probar que se inserte una categoría al sistema	
<b>Condiciones de Ejecución:</b> Todos los datos son insertados correctamente y que la sesión esté iniciada por el administrador.	
<b>Entrada / Pasos de ejecución:</b> Se intenta guardar la categoría en la base de datos.	
<b>Resultado Esperado:</b> La categoría es insertada correctamente y guardada en la base de datos del sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 70 Prueba de aceptación para la historia de usuario Gestionar Categoría.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU4_P2	<b>Historia de Usuario:</b> 4 – Gestionar Categoría
<b>Nombre:</b> Modificar una categoría existente en el sistema.	
<b>Descripción:</b> Se intenta modificar una categoría existente en el sistema.	

<b>Condiciones de Ejecución:</b> Todos los datos son insertados correctamente y que la sesión esté iniciada por el administrador.
<b>Entrada / Pasos de ejecución:</b> Se intenta guardar los nuevos datos de la categoría en la base de datos.
<b>Resultado Esperado:</b> Los nuevos datos son guardados correctamente en la base de datos del sistema.
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.

**Tabla 71 Prueba de aceptación para la historia de usuario Gestionar Categoría.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU4_P3	<b>Historia de Usuario:</b> 4 – Gestionar Categoría
<b>Nombre:</b> Eliminar una categoría del sistema	
<b>Descripción:</b> Se desea eliminar una categoría del sistema	
<b>Condiciones de Ejecución:</b> La sesión debe ser iniciada por el administrador.	
<b>Entrada / Pasos de ejecución:</b> Se intenta eliminar una categoría de la base de datos.	
<b>Resultado Esperado:</b> La categoría es eliminada correctamente de la base de datos del sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 72 Prueba de aceptación para la historia de usuario Gestionar Métrica.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU5_P1	<b>Historia de Usuario:</b> 5 – Gestionar Métrica
<b>Nombre:</b> Insertar una métrica al sistema	
<b>Descripción:</b> Probar que se inserte una métrica al sistema	
<b>Condiciones de Ejecución:</b> Todos los datos son insertados correctamente y que la sesión esté iniciada por el administrador.	
<b>Entrada / Pasos de ejecución:</b> Se intenta guardar la métrica en la base de datos.	

<b>Resultado Esperado:</b> La métrica es insertada correctamente y guardada en la base de datos del sistema.
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.

**Tabla 73 Prueba de aceptación para la historia de usuario Gestionar Métrica.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU5_P2	<b>Historia de Usuario:</b> 5 – Gestionar Métrica
<b>Nombre:</b> Modificar una métrica existente en el sistema.	
<b>Descripción:</b> Se intenta modificar una métrica existente en el sistema.	
<b>Condiciones de Ejecución:</b> Todos los datos son insertados correctamente y que la sesión esté iniciada por el administrador.	
<b>Entrada / Pasos de ejecución:</b> Se intenta guardar los nuevos datos de la métrica en la base de datos.	
<b>Resultado Esperado:</b> Los nuevos datos son guardados correctamente en la base de datos del sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 74 Prueba de aceptación para la historia de usuario Gestionar Métrica.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU5_P3	<b>Historia de Usuario:</b> 5 – Gestionar Métrica
<b>Nombre:</b> Eliminar una métrica del sistema	
<b>Descripción:</b> Se desea eliminar una métrica del sistema	
<b>Condiciones de Ejecución:</b> La sesión debe ser iniciada por el administrador.	
<b>Entrada / Pasos de ejecución:</b> Se intenta eliminar la métrica de la base de datos.	
<b>Resultado Esperado:</b> La métrica es eliminada correctamente de la base de datos del sistema.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 75 Prueba de aceptación para la historia de usuario Utilizar Métrica.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU6_P1	<b>Historia de Usuario:</b> 6 – Utilizar Métrica
<b>Nombre:</b> Utilizar métrica.	
<b>Descripción:</b> Probar que se puede utilizar una métrica existente en el sistema	
<b>Condiciones de Ejecución:</b> Todos los datos son insertados correctamente y que la sesión esté iniciada por el administrador o un líder de proyecto.	
<b>Entrada / Pasos de ejecución:</b> Se intenta calcular el resultado según la fórmula de la métrica escogida y los datos entrados por el usuario.	
<b>Resultado Esperado:</b> La utilización de la métrica arroja el resultado esperado.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 76 Prueba de aceptación para la historia de usuario Obtener Reporte.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU7_P1	<b>Historia de Usuario:</b> 7 – Obtener Reporte
<b>Nombre:</b> Obtener Reporte	
<b>Descripción:</b> Obtener un reporte de los últimos resultados arrojados por una métrica, esto ocurre cada vez que esta es utilizada.	
<b>Condiciones de Ejecución:</b> Se evalúa la métrica seleccionada y la sesión debe estar iniciada por el administrador o el líder de proyecto.	
<b>Entrada / Pasos de ejecución:</b> Se buscan en la base de datos todos los resultados asociados a la métrica.	
<b>Resultado Esperado:</b> El usuario puede ver todos los resultados de la métrica que acaba de evaluar.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 77 Prueba de aceptación para la historia de usuario Asignar Proyecto.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU8_P1	<b>Historia de Usuario:</b> 8 – Asignar Proyecto
<b>Nombre:</b> Asignar Proyecto	

<b>Descripción:</b> Se trata de asignar un proyecto a un líder que no tenga ninguno asociado
<b>Condiciones de Ejecución:</b> La sesión de estar iniciada por el administrador.
<b>Entrada / Pasos de ejecución:</b> Se intenta crear la relación líder---proyecto.
<b>Resultado Esperado:</b> El proyecto ha sido relacionado con el líder y este último puede hacer uso de las métricas para su posterior evaluación.
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.

**Tabla 78 Prueba de aceptación para la historia de usuario Obtener Comportamiento por Métrica.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU9_P1	<b>Historia de Usuario:</b> 9 – Obtener Comportamiento por Métrica.
<b>Nombre:</b> Obtener Comportamiento por Métrica.	
<b>Descripción:</b> Obtener el Comportamiento dada una métrica determinada.	
<b>Condiciones de Ejecución:</b> La sesión debe estar iniciada por el Líder de proyecto, el cual solo podrá acceder a las métricas utilizadas por su proyecto y a los resultados arrojados por este.	
<b>Entrada / Pasos de ejecución:</b> Se buscan en la base de datos todos los resultados asociados a la métrica escogida y en dependencia del proyecto que este asociado al líder.	
<b>Resultado Esperado:</b> El usuario puede ver todos los resultados de la métrica que escogió.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Tabla 79 Prueba de aceptación para la historia de usuario Obtener Historial de proyecto por Métrica.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU9_P1	<b>Historia de Usuario:</b> 9 – Obtener historial de proyecto por Métrica.
<b>Nombre:</b> Obtener historial de proyecto por Métrica.	
<b>Descripción:</b> Obtener el historial dado un proyecto y una métrica determinada.	

<b>Condiciones de Ejecución:</b> La sesión debe estar iniciada por el Administrador, el cual podrá ver todos los proyectos existentes en la universidad y tendrá acceso a cada uno de los historiales de las métricas utilizadas por estos.
<b>Entrada / Pasos de ejecución:</b> Se buscan en la base de datos todos los resultados asociados a la métrica y el proyecto escogidos.
<b>Resultado Esperado:</b> El usuario puede ver todos los resultados de la métrica de acuerdo al proyecto que seleccionó.
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.

#### 4.5 Conclusiones del Capítulo.

En este capítulo se abordaron las fases de implementación y pruebas del software desarrollado mostrando todos los artefactos generados a través del proceso de desarrollo, así como las pruebas realizadas al software durante el desarrollo del mismo.

## CAPITULO 5: ESTUDIO DE FACTIBILIDAD

### 5.1 Introducción

El análisis del costo, el esfuerzo y los beneficios es de suma importancia a la hora de implementar un sistema. En este capítulo se detalla el método estimación aplicado al sistema propuesto.

### 5.2 Características del proyecto

**Tabla 80 Entradas externas.**

Nombre de la Entrada Externa.	Cantidad de Ficheros.	Cantidad de Elementos de Datos.	Clasificación(Simple, Media o Compleja)
Gestionar Proyecto	1	5	Simple
Gestionar Categoría	1	5	Simple
Gestionar Métrica	1	6	Simple
Gestionar Resultado	1	2	Simple
<b>TOTAL</b>		<b>18</b>	

**Tabla 81 Salidas externas.**

Nombre de la salida externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (simple, media y compleja)
Obtener Reporte	1	3	Simple
Obtener Historial de Proyecto por Métrica	1	3	Simple
Obtener comportamiento por Métrica	1	3	Simple
<b>TOTAL</b>		<b>9</b>	

**Tabla 82 Consultas Externas.**

Nombre de la	Cantidad de	Cantidad de	Clasificación (simple, media

<b>consulta externa</b>	<b>ficheros</b>	<b>elementos de datos</b>	<b>y compleja)</b>
Asignar Proyecto	1	3	Simple
Gestionar Proyecto	1	3	Simple
Gestionar Categoría	1	3	Simple
Gestionar Métrica	1	3	Simple
Gestionar Resultado	1	3	Simple
<b>Total</b>			<b>15</b>

**Tabla 83 Archivos lógicos internos.**

<b>Nombre del fichero interno</b>	<b>Cantidad de ficheros</b>	<b>Cantidad de elementos de datos</b>	<b>Clasificación (simple, media y compleja)</b>
Proyecto	1	1	Simple
Categoría	1	1	Simple
Métrica	1	1	Simple
Resultado	1	1	Simple
<b>Total</b>		<b>4</b>	

**Tabla 84 Puntos de función desajustados.**

Elementos	Simple		Medio		Complejo		Subtotal
	No.	Peso	No.	Peso	No.	Peso	
Entradas externas	18	3	0	4	0	6	54
Salidas externas	9	4	0	5	0	7	36
Consultas Externas	15	3	0	4	0	6	45
Archivos L. internos	4	7	0	10	0	15	28
<b>TOTAL</b>							<b>163</b>

**5.3 Cálculo de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.**

El modelo COCOMO II usa Puntos Función y/o Líneas de Código Fuente (SLOC) como base para medir tamaño en los modelos de estimación de Diseño Temprano y Post-Arquitectura. Los Puntos de Función procuran cuantificar la funcionalidad de un sistema de software. La meta es obtener un número que caracterice completamente al sistema. Son útiles estimadores ya que están basados en información que está disponible en las etapas tempranas del ciclo de vida del desarrollo de *software*. COCOMO II considera solamente UFP (del inglés: Puntos Función Desajustados). [8]

**Tabla 85 Características.**

<b>Características</b>	<b>Valor</b>
Puntos de función desajustados	163
Lenguaje (php)	53
Instrucciones fuentes por puntos de función	8639
Instrucciones fuentes	8.639 ksloc

**Tabla 86 Factores de escala.**

<b>Nombre</b>	<b>Valor</b>	<b>Justificación</b>
PREC	1.24	Existen varias aplicaciones similares a nivel mundial, pero no a nivel nacional.
FLEX	2.03	Cuenta con una alta flexibilidad.
TEAM	1.10	El equipo de desarrollo presenta muy alta cohesión y madurez.
RESL	2.83	Se han identificado muy pocos riesgos
PMAT	3.12	Existe una experiencia previa en el desarrollo de aplicaciones web.
<b>Total(FE)</b>	<b>10.32</b>	

**Tabla 87 Multiplicadores de esfuerzo.**

<b>Nombre</b>	<b>Valor</b>	<b>Justificación</b>
<b>DOCU</b>	1.00	Documentación necesaria para el desarrollo del proyecto
<b>RELY</b>	1.00	Una falla en el software tendría un impacto medio.
<b>RUSE</b>	1.07	El nivel de reusabilidad del código es alto.
<b>ACAP</b>	0.85	La capacidad de análisis y diseño, eficiencia, habilidad para comunicarse y trabajar en equipo es alta.

<b>AEXP</b>	1.00	Nivel de experiencia del grupo de desarrollo en aplicaciones equivalentes es normal.
<b>PEXP</b>	0.91	Experiencia del grupo de desarrollo (principalmente programadores) en el uso de herramientas de software y hardware utilizado como plataforma.
<b>LTEX</b>	0.91	Experiencia del grupo de desarrollo en el lenguaje de programación y las herramientas de desarrollo utilizadas.
<b>PCAP</b>	0.88	La capacidad de trabajo en equipo, eficiencia y habilidad para comunicarse es alta.
<b>Total(ME)</b>	<b>0.66</b>	

**Cálculos**

**A = 2.94 C = 3.67 D = 0.24  $\Sigma$  (FE) = 10.32  $\Pi$  (ME) = 0.66 Sal = 349**

**Size = 53 x 163 = 8639 SLOC = 8.639 KSLOC**

**B = 0.91 + 0.01 x  $\Sigma$  (FE) = 0.91 + 0.01 x 10.32 = 1.0132**

**PMnominal = A x Size ^B = 2.94 x 8.639^1.0132 = 26.13**

**PMajustado = PMnominal x  $\Pi$  (ME) = 26.13 x 0.66 = 17.25**

**F = D + 0.2 x 0.01 x  $\Sigma$  (FE) = 0.24 + 0.2 x 0.01 x 10.32 = 0.26**

**TDEV = C x PMajustado^F = 3.67 x 17.25^0.26 = 7.70**

**CH = PMajustado/ TDEV = 17.25 / 7.70 = 2.24  $\approx$  2**

**Co. = CH x Sal x PMajustado = 2 x 349 x 17.25 = \$ 12040**

**Tabla 88 Resultados.**

<b>Cálculo de:</b>	<b>Valor</b>
Esfuerzo	17.25 hombres/mes
Tiempo de desarrollo	7.70 meses
Cantidad de hombres	2 hombres
Salario medio	\$349
Costo	\$12040

#### **5.4 Beneficios tangibles e intangibles**

El desarrollo del sistema propuesto en este trabajo brinda beneficio principalmente al departamento de calidad de la Universidad de Ciencias Informáticas, pues le permite a cada uno de los líderes de proyecto tanto evaluar cómo hacer un seguimiento durante el ciclo de vida de los mismos, permitiendo economizar gran parte del tiempo que se dedicaba al trabajo realizado manualmente, además, permite a la dirección de calidad llevar un control sistemático de todas las evaluaciones realizadas a la diferentes proyectos productivos. El Sistema de Gestión y Control de Métricas no está concebido para ser un producto comercial, aunque en un futuro pueda reportar beneficios monetarios a la entidad adquiriendo este carácter.

### **5.5 Análisis de costo**

Cada producto que se desarrolle tiene un costo de producción asociado, el cual debe ser justificado en base a los beneficios reportados por el mismo. El sistema propuesto en este trabajo de diploma no lleva consigo grandes gastos, debido a que solo contribuye a este el salario de los desarrolladores, debido a esto se llega a la conclusión de que su implementación es factible, a esto está muy vinculado la utilización de plataformas y herramientas libres que no requieren la compra de licencia ni ningún otro tipo de pago asociado a ello.

### **5.6 Conclusiones del Capítulo.**

En este capítulo se realizó un análisis de factibilidad de la solución propuesta, llegando a la conclusión de que es viable su desarrollo comparando los costos de producción con los beneficios reportados por su puesta en funcionamiento.

## **CONCLUSIONES GENERALES**

Con la realización de este trabajo se arribaron a las siguientes conclusiones:

El intercambio constante con los clientes y la aplicación de estrategias para la captura de requisitos, facilitó entender el negocio actual y describirlo, para realizar los artefactos correspondientes, y la obtención de la información necesaria para delimitar las funcionalidades, conformando así los requisitos del sistema.

A partir del estudio realizado de las metodologías de desarrollo de software, se escogió XP como la más adecuada para el desarrollo de este trabajo. Dicha metodología y las herramientas definidas permitieron que el proceso de desarrollo se realizara con una mayor organización y calidad.

La aplicación de las pruebas al sistema garantizó una mayor fiabilidad y utilidad del mismo, además permitió que todas las funcionalidades que se especificaron al inicio por parte de los clientes fueran cumplidas satisfactoriamente, contribuyendo así al logro de la calidad del producto final en el tiempo requerido y con un mínimo de los costos.

Durante el transcurso del presente trabajo se logró afianzar los conocimientos adquiridos a lo largo de la carrera y posibilitó obtener otros que fueron necesarios para el desarrollo del mismo.

## RECOMENDACIONES

Al finalizar este trabajo quedan algunas recomendaciones que pueden servir de punto de partida para mejorar aún más el sistema obtenido:

- ❖ Continuar con las investigaciones para añadir nuevas funcionalidades al sistema y obtener mejoras en futuras versiones, logrando adecuarlo cada vez más a las necesidades de la Dirección Central de Calidad.
- ❖ Publicar el sistema, dándole la posibilidad a los usuarios del dominio UCI.CU de conectarse a este, de forma tal que estos aprendan más sobre métricas y sus usos y de igual forma aporten ideas de los posibles cambios o mejoras a desarrollar en la aplicación.

---

## BIBLIOGRAFÍA

1. **Anaya Villegas, Adrian.** Monografías. *Monografías*. [En línea] Mayo de 2007. [Citado el: 09 de Abril de 2008.] <http://www.monografias.com/trabajos51/programacion-extrema/programacion-extrema.shtml>.
2. **Cameron, Nadia.** ComputerWorld the voice of it management. *ComputerWorld the voice of it management*. [En línea] 20 de Enero de 2003. [Citado el: 10 de Enero de 2008.] <http://www.computerworld.com.au/index.php?id=760310963>.
3. **Cano Martínez, Rafael German.** Univalle. *Univalle*. [En línea] Septiembre de 2007. [Citado el: 11 de Abril de 2008.] <http://eisc.univalle.edu.co/~pais/contenidos/anteproyectosEnEvaluacion/RafaelCano.pdf>.
4. **Canós, José H.** xelphos. *xelphos*. [En línea] [Citado el: 13 de Abril de 2008.] [http://www.xelphos.com.ar/xop\\_ma\\_xp.html#](http://www.xelphos.com.ar/xop_ma_xp.html#).
5. **Dondo, Agustín.** PHP en Castellano. *PHP en Castellano*. [En línea] 03 de Febrero de 2005. [Citado el: 15 de Enero de 2008.] <http://www.programacion.com/php/articulo/porquephp/>.
6. **Fahnle, Pablo.** ASP en Castellano. *ASP en Castellano*. [En línea] Diciembre de 2000. [Citado el: 18 de Enero de 2008.] [http://www.programacion.com/asp/articulo/aspnet\\_queues/](http://www.programacion.com/asp/articulo/aspnet_queues/).
7. **Escribano, Gerardo Fernández.** Departamento de Sistemas Informáticos. *Departamento de Sistemas Informáticos*. [En línea] 2002. [Citado el: 15 de Abril de 2008.] <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-XP.pdf>.
8. **Letelier, Patricio.** willydev. *willydev*. [En línea] [Citado el: 15 de Abril de 2008.] <http://www.willydev.net/descargas/masyxp.pdf>.
9. **Masip, David.** desarrollo web.com. *desarrollo web.com*. [En línea] 2006. [Citado el: 10 de Enero de 2008.] <http://www.desarrolloweb.com/articulos/840.php>.
10. *ESTIMACIÓN DEL ESFUERZO BASADA EN CASOS DE USO.* **Peralta, Mario.** Buenos Aires : s.n.
11. **PostgreSQL, El equipo de desarrollo de.** TLDP-es. *TLDP-es*. [En línea] 2002. [Citado el: 21 de Enero de 2008.] <http://es.tldp.org/Postgresql-es/web/navegable/tutorial/tutorial.html>.
12. **Ramírez, Elsa.** SG Software Guru. *SG Software Guru*. [En línea] 06 de Marzo de 2008. [Citado el: 15 de Abril de 2008.] <http://www.sg.com.mx/content/view/684/1/>.
13. **Mendoza Sanchez, María A.** Informatizate. *Informatizate*. [En línea] 07 de Junio de 2004. [Citado el: 25 de Marzo de 2008.] [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).

14. **Suehring, Steve.** *PERL. Introduccion al desarrollo Web.* España : Anaya, Grupo Editorial, 2006.
  15. **Gonzales Urmachea, Mabel.** Monografías. *Monografías.* [En línea] 26 de Junio de 2006. [Citado el: 15 de Diciembre de 2007.] <http://www.monografias.com/trabajos14/sqlserver/sqlserver.shtml>.
  16. **Valle, José Guillermo.** Monografías. *Monografías.* [En línea] 2005. [Citado el: 25 de Enero de 2008.] <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>.
  17. **Gallego Vázquez, José Antonio.** *Desarrollo Web con PHP y MySQL.* s.l. : Anaya Multimedia, 2003.
  18. **Wells, Don.** ExtremmeProgramming. *ExtremmeProgramming.* [En línea] 17 de Febrero de 2006. [Citado el: 16 de Abril de 2008.] <http://www.extremeprogramming.org/recentchanges.html>.
-

---

## REFERENCIAS BIBLIOGRÁFICAS

- [1]. **Anaya Villegas, Adrian.** Monografías. *Monografías*. [En línea] Mayo de 2007. [Citado el: 09 de Abril de 2008.] <http://www.monografias.com/trabajos51/programacion-extrema/programacion-extrema.shtml>.
- [2]. **Cameron, Nadia.** ComputerWorld the voice of it management. *ComputerWorld the voice of it management*. [En línea] 20 de Enero de 2003. [Citado el: 10 de Enero de 2008.] <http://www.computerworld.com.au/index.php?id=760310963>.
- [3]. **Canós, José H.** xelphos. *xelphos*. [En línea] [Citado el: 13 de Abril de 2008.] [http://www.xelphos.com.ar/xop\\_ma\\_xp.html#](http://www.xelphos.com.ar/xop_ma_xp.html#).
- [4]. **Dondo, Agustín.** PHP en Castellano. *PHP en Castellano*. [En línea] 03 de Febrero de 2005. [Citado el: 15 de Enero de 2008.] <http://www.programacion.com/php/articulo/porquephp/>.
- [5]. **Fahle, Pablo.** ASP en Castellano. *ASP en Castellano*. [En línea] Diciembre de 2000. [Citado el: 18 de Enero de 2008.] [http://www.programacion.com/asp/articulo/aspnet\\_quees/](http://www.programacion.com/asp/articulo/aspnet_quees/).
- [6]. **Letelier, Patricio.** willydev. *willydev*. [En línea] [Citado el: 15 de Abril de 2008.] <http://www.willydev.net/descargas/masyxp.pdf>.
- [7]. **Masip, David.** desarrollo web.com. *desarrollo web.com*. [En línea] 2006. [Citado el: 10 de Enero de 2008.] <http://www.desarrolloweb.com/articulos/840.php>.
- [8]. *ESTIMACIÓN DEL ESFUERZO BASADA EN CASOS DE USO*. **Peralta, Mario.** Buenos Aires : s.n.
- [9]. **PostgreSQL, El equipo de desarrollo de.** TLDP-es. *TLDP-es*. [En línea] 2002. [Citado el: 21 de Enero de 2008.] <http://es.tldp.org/Postgresql-es/web/navigable/tutorial/tutorial.html>.
- [10]. **Ramírez, Elsa.** SG Software Guru. *SG Software Guru*. [En línea] 06 de Marzo de 2008. [Citado el: 15 de Abril de 2008.] <http://www.sg.com.mx/content/view/684/1/>.
- [11]. **Mendoza Sanchez, María A.** Informatizate. *Informatizate*. [En línea] 07 de Junio de 2004. [Citado el: 25 de Marzo de 2008.] [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).
- [12]. **Suehring, Steve.** PERL. *Introduccion al desarrollo Web*. España : Anaya, Grupo Editorial, 2006.
- [13]. **Gonzales Urmachea, Mabel.** Monografías. *Monografías*. [En línea] 26 de Junio de 2006. [Citado el: 15 de Diciembre de 2007.] <http://www.monografias.com/trabajos14/sqlserver/sqlserver.shtml>.
- [14]. **Valle, José Guillermo.** Monografías. *Monografías*. [En línea] 2005. [Citado el: 25 de Enero de 2008.] <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>.
-

- [15]. **Gallego Vázquez, José Antonio.** *Desarrollo Web con PHP y MySQL.* s.l. : Anaya Multimedia, 2003.

## GLOSARIO DE TÉRMINOS

**UCI:** Universidad de las Ciencias Informáticas.

**XP:** Extreme Programming (Programación Extrema) una de las metodologías de desarrollo de software más exitosas en la actualidad.

**UML:** Unified Modeling Language (Lenguaje Unificado de Modelado) es la herramienta usada en la descripción y construcción de software reconocida por la industria como estándar.

**RUP:** Rational Unified Process (Proceso Unificado de Desarrollo) es la metodología de IBM Rational para el desarrollo y construcción de software basado íntegramente en UML como soporte a la metodología.

**PHP:** Acrónimo de Hypertext Preprocessor es un lenguaje interpretado de alto nivel, embebido en páginas HTML

**HTML:** Acrónimo del inglés Hipertext Markup Language (Lenguaje de marcas hipertextuales) es un lenguaje de marcado diseñado para estructurar texto y presentarlo en forma de hipertexto, que es el formato estándar de las páginas web.

**ASP:** Active Server Pages, lenguaje de programación, se trata principalmente de código C# embebido en páginas HTML

**Python:** Es un lenguaje de programación creado por Guido van Rossum en el año 1990.

**JavaScript:** Es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

**SGBD:** Sistema Gestor de Bases de Datos, es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista a la misma vez.

**MySQL:** Es un sistema de gestión de base de datos relacional.

**XML:** (Extensible Markup Language) Lenguaje de Marcado Extensible, desarrollado por el World Wide Web Consortium (W3C).

---

**CMS:** (Content Management System), Sistema de gestión de contenido.

**Editor WYSIWYG:** (What you see is what you get), lo que ves es lo que obtendrás, nos permiten editar el código de la página web tal y como se verá en el navegador.

**Módulos:** Los módulos en Drupal son ficheros con extensión .module que contienen funciones escritas en PHP.

**Artefactos:** Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

**Requisitos funcionales:** Se refieren a las funciones específicas del software y definen qué es lo que se espera que realice el producto software que se desarrollará.

**Requisito no Funcional:** Requisito que especifica propiedades del sistema, como restricciones del entorno o de implementación, rendimiento, dependencias de la plataforma, mantenibilidad, extensibilidad o fiabilidad. Requisito que especifica restricciones físicas sobre un requisito funcional.