

# Trabajo de Diploma

Para Optar por el Título de

## Ingeniero Informático

**Título: Aplicación Web para la Gestión de la  
Información de la Actividad Científica e Innovación  
Tecnológica en la VRIP del ISMMM.**

**Autor: Deysi Columbié Hernández.**

**Tutor: Ing. Eloy Rafael Jiménez Iglesias.**

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Instituto Superior Minero Metalúrgico de Moa para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 16 días del mes de Junio del año 2013.

Deysi Columbié Hernández  
Nombre completo del autor

\_\_\_\_\_  
Firma

Ing. Eloy Rafael Jiménez Iglesias  
Nombre completo del tutor

\_\_\_\_\_  
Firma

# De sentimientos

*“Y si alguno de vosotros tiene falta de sabiduría,  
pídala a Dios, el cual da a todos abundantemente y sin  
reproche, y le será dada.”*

***Santa Biblia, Carta al Apóstol Santiago 1:5***

*“Porque yo Jehová soy tu Dios, quien te sostiene de tu  
mano derecha, y te dice: No temas, yo te ayudo.”*

***Santa Biblia, Libro de Isaías 41:13***

# AGradecimientos

*A mi Padre eterno, por estar siempre para mí, ayudándome y bendiciéndome  
todo el tiempo.*

*A Xiomara, mi mamita amada, por darme la vida, por ayudarme a vivirla,  
y por creer en mí.*

*A mi abuelita Carmen Delia, por los ayunos, las oraciones y los detalles.  
A mis tíos Aldo, Roberto y Roger; a mis tías Anita, Madelin y Nancy; a mis  
primas adoradas Grether, Gretcheen y Elizabeth; y a mi abuelo Roberto, por  
haberme ayudado a crecer, a ser mejor persona.*

*A mi madrina Xiomarita, a Chester, y al resto de mi familia, por estar a mi  
lado todos estos años.*

*A Lisandra, Estela, Glaubis, Ciro, Eulicer, Dulce, Vicky, Néstor, Peter,  
Carlos, Tomás, Cristiano, Salvador, Alicia, Aurelio, Rocky, y especialmente  
a Yoandy, por su amistad incondicional.*

*A mis profesores Alayo, Marcos, Nadira, Ileana, Alina, Legrá, Yisel,  
Roiky, y Eliober por sus consejos, ayuda, y enseñanzas.*

*A mis compañeros de aula, a José, por su ayuda invaluable, a Yuris, a Roger,  
por los dulces; a Isa, Katherine, Dailín, Thaís, Dayana, y a las demás niñas  
de mi cuarto, por hacer más llevadera mi vida de becada.*

*A todo aquel que dio el frente cuando necesité una mano(o dos)... mil gracias.*

*A aquellos que me hirieron y lastimaron... también gracias, porque con cada  
herida me hice más fuerte.*

*Sin ustedes hoy yo no sería quien soy, y ninguno de mis sueños hubiese sido  
posible.*

# Dedicatoria

*A mi mamita amada...*

*A mi abuelita Carmen Delia...*

*A Grether, Gretcheen y Elizabeth...*

*A mi familia...*

*A Chester...*

*A mis amigos y hermanos...*

*A ti...*

### **Resumen**

El Instituto Superior Minero Metalúrgico de Moa, como parte del sistema educacional cubano se ha enfrascado en la tarea de la investigación científica y la innovación tecnológica, procesos que ayudan a un mejor desempeño de nuestra realidad revolucionaria.

En nuestro centro, el proceso de control y manipulación de la información relacionada con esta tarea ocupa un lugar primordial en las actividades del instituto. Actualmente, no se cuenta con una herramienta informática que facilite la organización, control y centralización de esta información.

Por esta razón, y debido a la utilización de las TIC, surge la necesidad de crear una aplicación informática que permita gestionar la información referente a los subprocesos de la ciencia e innovación tecnológica en el Instituto Superior Minero Metalúrgico de Moa.

La presente investigación tiene como objetivo primario la informatización de la gestión de la actividad científica y la innovación tecnológica a partir de la creación de una aplicación web que maneje la información de manera rápida y sencilla.

Para la realización de la misma se hizo un estudio sobre las principales tecnologías y herramientas para la construcción de aplicaciones web. En el este documento se recoge la investigación efectuada así como la metodología de desarrollo de software seleccionada para darle solución a la problemática encontrada.

Abstract

This investigation, is in pursuit to contribute to the correct administration of the information generated by the process of science and tech innovation in The Mining Metallurgist Superior Institute of Moa.

The Mining Metallurgist Superior Institute of Moa, like part of the Cuban educational system is focused in the task of the scientific investigation and tech innovation, processes that help to a better acting of our revolutionary reality.

In our center, the process of control and manipulation of the information related with this task occupies a primordial place in the activities of the institute. Now, there's not an informatics tool that facilitates the organization, control and centralization of that information.

For this reason, and due to the utilization of the CIT, it surges the necessity of creating an informatics application that allows administering the information about the processes of the science and tech innovation in The Mining Metallurgist Superior Institute of Moa.

This investigation has the informatization of the administration of the scientific activity starting from the creation of a web application that manages the information in a rapid and simple manner like primary objective.

For the realization of the project was carried out a study on the principal technologies and gear for the construction of web applications. Within this document, is picked up the investigation as well as the methodology of software development selected in order to give a proper solution to the opposing problem.

## ÍNDICE

|  |    |
|--|----|
| ÍNDICE DE FIGURAS .....  | 9  |
| ÍNDICE DE TABLAS.....  | 10 |
| INTRODUCCIÓN.....  | 15 |
| CAPÍTULO I - FUNDAMENTACIÓN TEÓRICA.....   | 19 |
| 1.1 Estado del Arte .....  | 19 |
| 1.1.1 Antecedentes .....   | 19 |
| 1.1.2 Sistema relacionados al campo de acción .....  | 19 |
| 1.1.3 Flujo Actual de Procesos en la VRIP .....  | 21 |
| 1.1.4 Propuesta de Solución .....  | 22 |
| 1.2 Tendencias y tecnologías actuales .....  | 22 |
| 1.2.1 Políticas de Migración a Software Libre .....  | 22 |
| 1.2.2 Aplicaciones Web .....   | 23 |
| 1.2.3 Ventajas de una Web .....  | 24 |
| 1.2.4 Lenguajes de programación .....  | 25 |
| 1.2.5 Sistemas de gestión de bases de datos .....  | 27 |
| 1.2.6 Servidores para aplicaciones web .....   | 29 |
| 1.2.7 Paquete de tecnología XAMPP .....  | 30 |
| 1.3 Metodologías de desarrollo de software .....   | 31 |
| 1.3.1 Metodología propuesta para el desarrollo de la aplicación.....                       | 31 |
| 1.3.2 Proceso Unificado de Desarrollo (RUP) .....  | 32 |
| 1.3.3 Programación Extrema (Extreme Programming, XP) .....                                 | 33 |
| 1.3.4 Fundamentación de la selección de la metodología de desarrollo de software.<br>..... | 40 |
| 1.4 Framework de desarrollo web .....  | 41 |
| 1.4.1 Patrón MVC .....   | 41 |
| 1.4.2 Tipos de Framework WEB.....  | 43 |
| 1.4.3 Características .....  | 45 |
| 1.4.4 CodeIgniter .....  | 46 |
| 1.4.5 Zend Framework .....   | 47 |
| 1.4.6 Yii Framework.....   | 49 |
| 1.4.7 Fundamentación de la selección del Framework Yii.....                                | 54 |
| 1.4.8 Herramientas CASE .....  | 55 |
| 1.5 Herramientas a utilizar en la propuesta de solución .....                              | 55 |
| Conclusiones del Capítulo .....  | 57 |
| CAPÍTULO II – PLANEACIÓN Y DISEÑO.....   | 58 |
| 2.1 Funcionalidades generales .....  | 58 |
| 2.1.1 Personal relacionado con el sistema .....  | 58 |
| 2.1.2 Lista de reserva .....   | 59 |
| 2.1.3 Características del Sistema.....   | 60 |
| 2.1.4 Historias de usuario.....  | 61 |
| 2.2 Planificación de entregas .....  | 62 |
| 2.2.1 Estimación de esfuerzo por historias de usuario .....                                | 63 |



|  |       |
|--|-------|
| 2.2.2 Planificación de iteraciones .....           | 64    |
| 2.2.3 Plan de duración de las iteraciones.....     | 65    |
| 2.3 Clases, responsabilidades y colaboración ..... | 66    |
| Conclusiones del Capítulo .....                    | 67    |
| CAPÍTULO III – DESARROLLO Y PRUEBAS.....           | 68    |
| 3.1 Modelo de datos .....                          | 68    |
| 3.3 Desarrollo de las iteraciones.....             | 69    |
| 3.4 Tareas por historias de usuario.....           | 69    |
| 3.5 Pruebas.....                                   | 72    |
| 3.5.1 Pruebas de aceptación.....                   | 73    |
| Conclusiones del Capítulo .....                    | 75    |
| CAPÍTULO IV - ESTUDIO DE FACTIBILIDAD .....        | 76    |
| 4.1 Introducción.....                              | 76    |
| 4.2 Efectos Económicos .....                       | 76    |
| 4.3 Ficha de Costo .....                           | 78    |
| Conclusiones del Capítulo .....                    | 81    |
| CONCLUSIONES GENERALES .....                       | 82    |
| RECOMENDACIONES DEL AUTOR .....                    | 83    |
| REFERENCIAS BIBLIOGRÁFICAS .....                   | 84    |
| BIBLIOGRAFÍAS .....                                | 86    |
| GLOSARIO DE TÉRMINOS .....                         | 88    |
| Anexo 1 Historias de Usuario .....                 | I     |
| Anexo 2 Tarjetas CRC.....                          | X     |
| Anexo 3 Tareas de Ingeniería .....                 | XVI   |
| Anexo 4 Pruebas de Aceptación .....                | XXXV  |
| Anexo 5 Interfaz de Usuario .....                  | XLVII |

## **ÍNDICE DE FIGURAS**

|   |        |
|---|--------|
| Figura 1 Modelo Vista Controlador .....                                   | 42     |
| Figura 2 Frontend - Backend .....   | 44     |
| Figura 3 Funcionamiento de Yii Framework .....                            | 51     |
| Figura 4 Modelo de datos .....  | 69     |
| Figura 5 Gráfico relación solución manual y solución con el software..... | 80     |
| Figura 6 Interfaz de usuario Insertar proyecto de investigación .....     | XLVII  |
| Figura 7 Interfaz de usuario insertar documento .....                     | XLVIII |
| Figura 8 Interfaz de usuario Insertar usuario .....                       | XLIX   |
| Figura 9 Interfaz de usuario Insertar miembro de proyecto. ....           | XLIX   |

## **ÍNDICE DE TABLAS**

|            |   |     |
|------------|---|-----|
| Tabla 2.1  | Personas relacionadas con el sistema .....                          | 58  |
| Tabla 2.2  | Historia de Usuario No.1 .....                                      | 62  |
| Tabla 2.3  | Esfuerzo por HU .....   | 63  |
| Tabla 2.4  | Plan de duración de las iteraciones .....                           | 65  |
| Tabla 2.5  | Tarjetas CRC Gestionar indicadores dinámicos de la evaluación. .... | 66  |
| Tabla 3.1  | Distribución de tareas de ingeniería por historia de usuario .....  | 70  |
| Tabla 3.2  | Historias abordadas en la primera iteración .....                   | 71  |
| Tabla 3.3  | Historias de usuario abordadas en la segunda iteración .....        | 72  |
| Tabla 3.4  | Historias de usuario abordadas en la tercera iteración .....        | 72  |
| Tabla 3.5  | Prueba de aceptación .....  | 73  |
| Tabla 2.6  | HU No. 1 Autenticar Usuario .....                                   | I   |
| Tabla 2.7  | HU No. 2 Gestionar Usuario .....                                    | I   |
| Tabla 2.8  | HU No. 3 Gestionar Línea de Investigación .....                     | II  |
| Tabla 2.9  | HU No. 4 Gestionar indicadores dinámicos de la evaluación .....     | II  |
| Tabla 2.10 | HU No. 5 Gestionar proyectos .....                                  | III |
| Tabla 2.11 | HU No. 6 Gestionar secuencia de actividades. ....                   | III |
| Tabla 2.12 | HU No. 7 Gestionar miembro del proyecto .....                       | IV  |
| Tabla 2.13 | HU No. 8 Gestionar documentos .....                                 | IV  |
| Tabla 2.14 | HU No. 9 Gestionar plan de resultados .....                         | V   |
| Tabla 2.15 | HU No. 10 Registrar software .....                                  | V   |
| Tabla 2.16 | HU No. 11 Listar registros de software .....                        | VI  |
| Tabla 2.17 | HU No. 12 Gestionar artículo .....                                  | VI  |
| Tabla 2.18 | HU No. 13 Gestionar evento .....                                    | VII |

|  |       |
|--|-------|
| Tabla 2.19 HU No. 14 Gestionar premios.....                            | VII   |
| Tabla 2.19 HU No. 15 Registrar publicación.....                        | VIII  |
| Tabla 2.20 HU No. 16 Listar registro de publicación. ....              | VIII  |
| Tabla 2.21 Tarjeta CRC No. 1 .....                                     | X     |
| Tabla 2.22 Tarjeta CRC No. 2 .....                                     | X     |
| Tabla 2.23 Tarjeta CRC No. 4 .....                                     | X     |
| Tabla 2.24 Tarjeta CRC No. 5 .....                                     | XI    |
| Tabla 2.25 Tarjeta CRC No. 6 .....                                     | XI    |
| Tabla 2.26 Tarjeta CRC No. 7 .....                                     | XII   |
| Tabla 2.27 Tarjeta CRC No. 8 .....                                     | XII   |
| Tabla 2.28 Tarjeta CRC No. 9 .....                                     | XII   |
| Tabla 2.29 Tarjeta CRC No. 10 .....                                    | XIII  |
| Tabla 2.30 Tarjeta CRC No. 11 .....                                    | XIII  |
| Tabla 2.31 Tarjeta CRC No. 12 .....                                    | XIV   |
| Tabla 2.32 Tarjeta CRC No. 13 .....                                    | XIV   |
| Tabla 2.33 Tarjeta CRC No. 14 .....                                    | XIV   |
| Tabla 3.6 Tarea de programación Autenticar usuario.....                | XVI   |
| Tabla 3.7 Tarea de programación Insertar usuario.....                  | XVI   |
| Tabla 3.8 Tarea de programación Mostrar usuario.....                   | XVI   |
| Tabla 3.9 Tarea de programación Modificar usuario. ....                | XVII  |
| Tabla 3.10 Tarea de programación Eliminar usuario.....                 | XVII  |
| Tabla 3.11 Tarea de programación Insertar línea de investigación. .... | XVII  |
| Tabla 3.12 Tarea de programación Mostrar línea de investigación. ....  | XVIII |
| Tabla 3.13 Tarea de programación Modificar línea de investigación..... | XVIII |

|   |       |
|---|-------|
| Tabla 3.14 Tarea de programación Eliminar línea de investigación .....              | XIX   |
| Tabla 3.15 Tarea de programación Insertar indicadores dinámicos de la evaluación. . | XIX   |
| Tabla 3.16 Tarea de programación Mostrar indicadores dinámicos de la evaluación. .  | XIX   |
| Tabla 3.17 Tarea de programación Modificar indicadores dinámicos de la evaluación.  | XX    |
| Tabla 3.18 Tarea de programación Eliminar indicadores dinámicos de la evaluación. . | XX    |
| Tabla 3.19 Tarea de programación Insertar proyectos .....                           | XX    |
| Tabla 3.20 Tarea de programación Mostrar proyectos .....                            | XXI   |
| Tabla 3.21 Tarea de programación Modificar proyectos .....                          | XXI   |
| Tabla 3.22 Tarea de programación Eliminar proyectos .....                           | XXII  |
| Tabla 3.23 Tarea de programación Insertar secuencia de actividades .....            | XXII  |
| Tabla 3.24 Tarea de programación mostrar secuencia de actividades .....             | XXII  |
| Tabla 3.25 Tarea de programación modificar secuencia de actividades .....           | XXIII |
| Tabla 3.26 Tarea de programación eliminar secuencia de actividades .....            | XXIII |
| Tabla 3.27 Tarea de programación insertar miembro de proyecto .....                 | XXIII |
| Tabla 3.28 Tarea de programación mostrar miembro de proyecto.....                   | XXIV  |
| Tabla 3.29 Tarea de programación Modificar miembro de proyectos.....                | XXIV  |
| Tabla 3.30 Tarea de programación Eliminar miembro de proyectos .....                | XXV   |
| Tabla 3.31 Tarea de programación Insertar documentos .....                          | XXV   |
| Tabla 3.32 Tarea de programación Mostrar documentos .....                           | XXV   |
| Tabla 3.33 Tarea de programación Modificar documento .....                          | XXVI  |
| Tabla 3.34 Tarea de programación Eliminar documento .....                           | XXVI  |
| Tabla 3.35 Tarea de programación Insertar plan de resultado .....                   | XXVI  |
| Tabla 3.36 Tarea de programación Mostrar plan de resultado .....                    | XXVII |
| Tabla 3.37 Tarea de programación Modificar plan de resultado.....                   | XXVII |

|  |        |
|--|--------|
| Tabla 3.38 Tarea de programación Eliminar plan de resultado .....                                    | XXVII  |
| Tabla 3.39 Tarea de programación Registrar software .....  | XXVIII |
| Tabla 3.40 Tarea de programación Listar registro de software .....                                   | XXVIII |
| Tabla 3.41 Tarea de programación Insertar artículo.....  | XXIX   |
| Tabla 3.41 Tarea de programación Mostrar un artículo.....  | XXIX   |
| Tabla 3.42 Tarea de programación Modificar un artículo .....   | XXIX   |
| Tabla 3.43 Tarea de programación Eliminar artículo .....   | XXX    |
| Tabla 3.44 Tarea de programación Insertar evento.....  | XXX    |
| Tabla 3.45 Tarea de programación Mostrar evento.....   | XXX    |
| Tabla 3.46 Tarea de programación Modificar evento .....  | XXXI   |
| Tabla 3.47 Tarea de programación Eliminar evento.....  | XXXI   |
| Tabla 3.54 Prueba para comprobar la entrada de un usuario al sistema .....                           | XXXV   |
| Tabla 3.55 Prueba para comprobar la entrada de datos de un usuario.....                              | XXXV   |
| Tabla 5.56: Prueba para comprobar la entrada de datos de la línea de investigación.<br>.....         | XXXVI  |
| Tabla 3.57 Prueba para comprobar la entrada de datos de indicadores dinámicos de<br>evaluación. .... | XXXVII |
| Tabla 3.58: Prueba para comprobar la entrada de datos del proyecto .....                             | XXXVII |
| Tabla 3.59: Prueba para comprobar la entrada de datos la secuencia de actividades<br>.....           | XXXIX  |
| Tabla 3.60: Prueba para comprobar la entrada de datos del miembro de proyecto                        | XXXIX  |
| Tabla 3.61: Prueba para comprobar la entrada de datos del documento.....                             | XLI    |
| Tabla 3.62: Prueba para comprobar la entrada de datos del plan de resultado. ....                    | XLI    |
| Tabla 3.63: Prueba para comprobar la entrada de datos de registro de software .....                  | XLIII  |
| Tabla 3.64: Prueba para comprobar la entrada de datos del artículo .....                             | XLIII  |
| Tabla 3.65: Prueba para comprobar la entrada de datos del evento.....                                | XLV    |

Tabla 3.66: Prueba para comprobar la entrada de datos del premio.....XLV

Tabla 6.67: Prueba para comprobar la entrada de datos del registro de publicación.XLVI

## **INTRODUCCIÓN**

El devenir histórico de la humanidad ha sido condicionado por los conocimientos que el hombre ha acumulado a través de los años, y su aplicación práctica a las situaciones cotidianas. En la actualidad, es común escuchar acerca de la importancia de la información, su relevancia entorno a todos los procesos de las organizaciones y entidades.

La información es un elemento fundamental para el desarrollo, y por consiguiente, la gestión de la información ha alcanzado un lugar cúspide en el avance de la economía, la medicina, la educación y todas las ramas posibles. Dentro de este ámbito la gestión de la información es toda la actividad relacionada con la obtención de la información adecuada, a un precio adecuado, en el tiempo y lugar adecuado, para tomar una decisión adecuada.

Nuestro país, a pesar de las limitantes encontradas en la continua lucha por crear una sociedad justa y que ponga los recursos en manos del pueblo, ha centrado esfuerzos en la investigación científica.

Nuestro centro, enmarcado en el ámbito del sistema educacional cubano aporta importantes investigaciones de profesores competentes, autoridades en las ramas de la geología y minería entre otras.

La Vicerrectoría de Investigación y Postgrado tiene a su cargo llevar el control de procesos vitales de esta actividad para el Ministerio de Educación Superior como son eventos científicos, publicaciones, concursos de carácter investigativo y de innovación tecnológica, además, controlar los proyectos de investigación que desarrollan los profesores y estudiantes del ISMMM. Esta información es necesario mantenerla organizada y accesible a las consultas. Para ello se recogen los datos de forma manual, y se guardan para su posterior uso.

Debido al gran volumen de información que se manipula en la VRIP y a la creciente necesidad de la utilización de las Tecnologías de la Información y las Comunicaciones el Instituto Superior Minero Metalúrgico de Moa (ISMMM), como parte del sistema



educacional cubano, se ha trazado como estrategia la informatización de la mayoría de sus procesos. El ISMMM tiene grandes problemas a la hora de consultar la información referente a la actividad científica que se realiza en el mismo, pues, actualmente no se cuenta con una vía que facilite y optimice la gestión de la información relacionada con los artículos, eventos, publicaciones, entre otras.

La **situación problémica** que se presenta es que se pierde información valiosa referente a la actividad científica realizada en el ISMMM, y se dificulta el acceso a la información a la hora de realizar consultas. A partir de esta situación se identifica el **problema a resolver** ¿Cómo favorecer la gestión de la información de la actividad científica e innovación tecnológica en la VRIP del ISMMM? En esta dirección surge esta investigación, la cual tiene como **objeto de estudio**: las aplicaciones informáticas que permitan la gestión de la información de la actividad científica y la innovación tecnológica, abarcando el **campo de acción** de la informatización de los procesos de la gestión de la información en la Vicerrectoría de Investigación del ISMMM.

Para darle solución al problema se traza como **objetivo general** la creación de una aplicación web que permita realizar la gestión de la información sobre la innovación tecnológica y la actividad científica, lo cual favorece el proceso de obtención de información en el Instituto Superior Minero Metalúrgico de Moa (ISMMM).

Para darle solución al objetivo general se trazan los siguientes **objetivos específicos**:

- Establecer el marco teórico de la investigación.
- Realizar un estudio de los procesos que constituyen la actividad científica y la innovación tecnológica en la Vicerrectoría de Investigación y Postgrado del ISMMM.
- Implementar la aplicación siguiendo la metodología de desarrollo de software seleccionada.
- Realizar el estudio de factibilidad del sistema.

Para darle cumplimiento a los objetivos específicos se definieron las siguientes **Tareas de investigación**:

1. Buscar los antecedentes de la investigación.
2. Realizar un estudio sobre las aplicaciones informáticas para la gestión de la información.
3. Seleccionar las tecnologías, herramientas y la metodología de desarrollo de software.
4. Estudiar el proceso de gestión de la información de la actividad científica e innovación tecnológica manejada en la Vicerrectoría de Investigación y Postgrado con los especialistas.
5. Realizar entrevistas a los especialistas.
6. Identificar los requerimientos del sistema.
7. Diseñar la base de datos.
8. Implementar la aplicación informática.
9. Realizar pruebas funcionales.
10. Realizar la evaluación costo-beneficio del sistema.

De esta forma la **idea a defender** se plantea en que si se elabora una aplicación web para la gestión de la información de la actividad científica y la innovación tecnológica, manejada en la Vicerrectoría de Investigación, se podrá favorecer al proceso de gestión de la información sobre los eventos, artículos, publicaciones, obtención de informes, para organizar la información así como también agilizar el proceso de trabajo.

El método empírico utilizado es:

**Entrevista:** para conocer cómo es el proceso de gestión de la información de la actividad científica y la innovación tecnológica en la VRIP, además para conocer los requerimientos que el sistema debe cumplir.

Los métodos teóricos utilizados son:

**Análisis y síntesis:** para la recopilación y el procesamiento de la información obtenida en el método empírico y arribar a las conclusiones de la investigación.

**Histórico – Lógico:** se puso de manifiesto durante el proceso de estudio de las herramientas a utilizar.

El presente trabajo de diploma estará estructurado en 4 capítulos:

**Capítulo 1: “Fundamentación Teórica”:** Este capítulo incluye un estado del arte del tema tratado, además se analizarán los diferentes lenguajes, tecnologías y metodologías de desarrollo de software empleadas en el desarrollo de la aplicación propuesta.

**Capítulo 2: “Análisis y Diseño”:** Se hace uso de la metodología expuesta en el capítulo anterior para el desarrollo del proyecto, abordando en detalles cada una de sus fases.

**Capítulo 3: “Implementación y Pruebas”:** Se presentan los principales métodos y definiciones dentro de la implementación de los flujos de trabajo. Se describen las pruebas realizadas y sus resultados.

**Capítulo 4: “Estudio de Factibilidad”:** En este se realiza un estudio para ver la factibilidad del producto por la metodología Coste-Beneficio. Además de un estudio de los esfuerzos requeridos para la realización del sistema propuesto.

## **CAPÍTULO I - FUNDAMENTACIÓN TEÓRICA**

En el presente capítulo se analizará todo el proceso de gestión de la información de la actividad científica e innovación tecnológica que se lleva a cabo en la Vicerrectoría de Investigación y Postgrado, además de la forma en que se archivan los documentos y otros informes relacionados. Se abordarán conceptos asociados al análisis de las tecnologías y herramientas a utilizar, además de la metodología seleccionada para el desarrollo del software. Se manifiesta una mejor visión sobre la herramienta a desarrollar. Este capítulo constituye la base teórica para la comprensión del trabajo que se desarrolla.

### **1.1 Estado del Arte**

#### **1.1.1 Antecedentes**

El continuo desarrollo de la tecnología a nivel mundial, ha facilitado que en muchos países se creen sistemas dedicados a la gestión de la información, la utilización de estos sistemas permite llevar un control más preciso de los datos, alcanzar mayores niveles en la centralización de la información, así como optimizar su disponibilidad. En la búsqueda de sistemas relacionados al que se pretende desarrollar se encontraron algunas herramientas que se dedican a la gestión de proyectos, entre ellos DotProject, el Redmine, entre otros, pero debido al alcance del sistema que se desea obtener, el cual, debe gestionar la información relacionada con las líneas de investigación, los eventos científicos y las publicaciones, entre otras; se determinó que era necesaria la creación de un sistema propio del instituto, que aglutinara estas funcionalidades en una sola herramienta.

#### **1.1.2 Sistema relacionados al campo de acción**

DotProject

Fue creado por la comunidad de DotProject, con el propósito de construir una herramienta para la Gestión de Proyectos. Está constituido por aplicaciones de código abierto. Es una aplicación web, multiusuario, soporta varios lenguajes y es software libre, además de ser multiplataforma. Está programada en PHP y utiliza

MySQL como base de datos. La plataforma recomendada para utilizar DotProject se denomina LAMP (Linux + Apache + MySQL + PHP).

- DotProject es una herramienta orientada a la Gestión de Proyectos. Para eso se orienta a la administración de recursos para desarrollar un producto, cuya producción requiera de un conjunto de actividades o tareas que se desarrollen entre ellas en forma paralela o independiente. La aplicación consta de un conjunto de entidades ordenadas jerárquicamente, las cuales permiten brindar la funcionalidad del producto. Algunas de estas son:
- Compañías: son las entidades que agrupan proyectos, actividades y usuarios.
- Departamentos: son áreas dentro de las compañías, que agrupan usuarios en dicho nivel.
- Usuarios: las personas registradas en DotProject que trabajan dentro del sistema como usuarios del mismo, de acuerdo al rol que le fue asignado.
- Contactos: son usuarios especiales que asignados a un determinado proyecto pueden recibir: correo, actualizaciones y noticias pero no necesariamente deben tener acceso al sistema DotProject. Los usuarios y contactos pertenecen a una compañía.
- Costos: Cuando se contrata a un consultor independiente para un proyecto, el costo típicamente será determinado por la tarifa de la empresa consultora multiplicada por un estimado del avance del proyecto.
- Proyectos: Es la entidad que contiene el grupo de tareas necesarias para desarrollar un determinado producto o servicio.
- Actividades: son las tareas asignadas dentro de un proyecto.
- Diagramas de Gantt: Permite ver en forma gráfica las actividades ordenadas jerárquicamente, mostrando las dependencias y solapamientos de las mismas.

(1)

### **RedMine**

RedMine es una de las web más populares basadas en la aplicación de la gestión de proyectos. Incluye gráficos de Gantt y calendario para ayudar a la

representación visual de los proyectos y sus plazos. Es compatible con múltiples proyectos y es multiplataforma. Gestor de proyectos de código libre para las empresas. Como base de datos soporta tanto MySQL como PostgreSQL o SQLite. La principal ventaja que aporta como gestor de proyectos es poder tener toda la información asociada a un proyecto acotada dentro del mismo, así como el avance del trabajo en comunidades de desarrollo. RedMine es una herramienta totalmente configurable. Aparte de crear múltiples proyectos y usuarios, permite modificar los valores de los campos y crear campos propios para adaptar la herramienta a cada cliente, crear roles con distintos permisos y crear flujos de trabajo personalizados. Además de permitir el control de la ejecución del proyecto, todo ello a través de una interfaz web que hace sencilla la gestión de los mismos. Es una aplicación de código abierto y liberado bajo los términos de la GNU General Public License (GPL). Dentro de sus características encontramos: (2)

- Soporta múltiples proyectos.
- Roles flexibles basados en control de acceso.
- Sistema de seguimiento de errores flexible.
- Diagramas de Gantt y calendario.
- Administración de noticias, documentos y archivos. Fuentes web y notificaciones por correo electrónico.

### **1.1.3 Flujo Actual de Procesos en la VRIP**

El Vicerrector de Investigación y Postgrado recibe las orientaciones del MES, donde se plantean los objetivos de trabajo del año en curso, las normas y resoluciones que regirán estos objetivos, entre otras. Luego, en la VRIP se elabora un informe del balance del año anterior, donde se analizan los subprocesos que constituyen la actividad científica e innovación tecnológica en nuestro instituto, y se adaptan los objetivos a las necesidades reales de nuestra institución, elaborándose los objetivos específicos para el año de trabajo.

Los factores que afectan el proceso de gestión de información en la VRIP han sido identificados como:

- Descentralización de la información.
- Dificultad en la obtención de información.
- Lentitud en las consultas a los datos.
- Acumulación de documentos impresos.
- Falta de un control de usuario para acceder a las informaciones.

#### **1.1.4 Propuesta de Solución**

Después de haber realizado el estudio y análisis de la problemática presentada se propone la creación de una aplicación web que permita realizar la gestión de la información relacionada con los procesos que constituyen la actividad científica e innovación tecnológica, como son la gestión de información sobre los eventos, artículos, publicaciones, entre otras, para favorecer el proceso de obtención de información en el ISMMM.

### **1.2 Tendencias y tecnologías actuales**

Ante el incesante avance de las tecnologías, la sociedad, ávida de nuevas herramientas y funcionalidades, exige a los desarrolladores de software nuevos retos y nuevas concepciones para satisfacer sus exigencias, cada vez más ambiciosas. La necesidad del conocimiento requiere cada vez más que la información esté disponible, que se conozca la importancia de esta, que se conozca del artículo científico en la investigación y en el aprendizaje. Además de la integración entorno al quehacer científico.

Para satisfacer estas exigencias, los desarrolladores deben buscar nuevas ideas surgiendo así nuevas metodologías y formas de desarrollo que permiten confeccionar productos cada vez más complejos.

#### **1.2.1 Políticas de Migración a Software Libre**

La globalización del mercado en el ámbito informático ha impactado grandemente en el tránsito de la economía dedicada a la producción de bienes y en el flujo de mercancías, impidiendo el desarrollo de las mismas. Debido a lo planteado anteriormente, las necesidades de desarrollo de las tecnologías están encaminadas a la realización de

productos que están orientados hacia la política de software libre, como es el caso del sistema que se implementará.

Se denomina software libre a todo aquel que permita a los usuarios ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. A menudo es confundido con el software gratuito, sin embargo no se trata de una cuestión de precio sino de libertad. Precisamente, las cuatro libertades que se definen son:

- ✓ La libertad de ejecutar el programa para cualquier propósito.
- ✓ La libertad de estudiar cómo trabaja el programa y adaptarlo a sus necesidades (El acceso al código fuente es una condición necesaria).
- ✓ La libertad de redistribuir copias para que pueda ayudar al vecino.
- ✓ La libertad de mejorar el programa y publicar sus mejoras y versiones modificadas en general para que se beneficie toda la comunidad (El acceso al código fuente es una condición necesaria). (3)

Las soluciones libres brindan ventajas, algunas están enfocadas principalmente en el aspecto económico, estas han permitido un constante crecimiento del software libre, en ocasiones, hasta superar al mercado propietario, como es el caso de los servidores web.

Al país le favorecen estas ventajas, por lo que tiende a la migración hacia software libre. Siguiendo este proceso se decide la utilización de herramientas y tecnologías pertenecientes a software libres para el desarrollo de la aplicación.

### **1.2.2 Aplicaciones Web**

Una aplicación web es un sistema informático que los usuarios usan accediendo a un servidor web a través de los protocolos de Internet. Las aplicaciones web son populares por lo práctico del navegador web como cliente ligero. La facilidad de actualizar y mantener las aplicaciones web sin distribuir e instalar software en miles de estaciones de trabajo es otra razón de su creciente popularidad. Aunque existen muchas variaciones posibles, una aplicación web está normalmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web ofrece la primera



capa y un motor capaz de usar alguna tecnología web dinámica (ejemplo: PHP, Java Servlets o ASP, ASP.NET, CGI, ColdFusion, embPerl, Python o Ruby on Rails) constituye la capa intermedia. Por último, una base de datos constituye la tercera y última capa. El navegador web manda peticiones a la capa intermedia que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario. (4)

### 1.2.3 Ventajas de una Web

- **Compatibilidad multiplataforma:** Las aplicaciones web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.
- **Actualización:** Las aplicaciones basadas en web están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo, pues no se hace necesario iniciar nuevas descargas y/o procedimientos de instalación (algunas veces imposible cuando usted está trabajando dentro de grandes organizaciones).
- **Inmediatez de acceso:** Las aplicaciones basadas en web no necesitan ser descargadas, instaladas y configuradas. Usted accede a su cuenta online y están listas para trabajar sin importar cuál es su configuración o su hardware.
- **Menos requerimientos de memoria:** Las aplicaciones basadas en web tienen menos demandas de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y correr en los servidores del proveedor, esas aplicaciones basadas en web usan en muchos casos la memoria de las computadoras donde ellas corren, dejando más espacio para correr múltiples aplicaciones sin incurrir en frustrantes deterioros en el rendimiento.
- **Menos Bugs:** Las aplicaciones basadas en web son menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con

aplicaciones basadas en web, todos utilizan la misma versión, y todos los bugs pueden ser corregidos tan pronto como son descubiertos.

- **Precio:** Las aplicaciones basadas en web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional. Esto permite que las aplicaciones online cuesten una fracción de sus contrapartes descargables y no totalmente gratuitas, mientras que ofrecen componentes adicionales y servicios Premium como una opción.
- **Múltiples usuarios concurrentes:** Las aplicaciones basadas en web pueden ser utilizadas por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta.
- **Los datos son más seguros:** Si bien la ruptura de discos no va a desaparecer, es probable que los usuarios escuchen mucho menos del tema. A medida que las compañías se hagan cargo del almacenamiento de los datos del usuario, granjas de almacenamiento de datos redundantes, altamente fiables, los usuarios van a tener mucho menos riesgo de perder sus datos debido a una ruptura de disco impredecible o a un virus de la computadora. Las compañías que proveen aplicaciones basadas en web van a brindar amplios servicios de resguardo de datos ya sea como una parte integral del servicio básico o como una opción paga. (5)

#### **1.2.4 Lenguajes de programación**

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en el conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente.

Pueden ser agrupados en dos grupos:

**Del lado del cliente:**

**Java Script:** Esto es un lenguaje interpretado por lo que no requiere compilación. Fue creado por la empresa Netscape Communication. Es similar al Java, aunque no es un lenguaje orientado a objetos (POO). La mayoría de los navegadores en sus últimas versiones interpretan JavaScript. El código de JavaScript puede ser integrado a nuestras páginas web.

**Del lado del servidor:**

**PHP:** Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-sidescriting) pero actualmente puede ser utilizado desde una interfaz de líneas de comando o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas QT (Quasar Toolkit). PHP es un acrónimo recursivo que significa Hypertext Pre-processor (inicialmente PHP Tools o Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994.  
(6)

**¿Por qué PHP?**

Ventajas de PHP

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de bases de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial en la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientadas a Objetos.
- Biblioteca nativa de funciones sumamente amplias e incluídas.
- No requiere definición de tipo de variables.

- Tiene manejo de excepciones. (7)

### **1.2.5 Sistemas de gestión de bases de datos**

Los Sistemas de Gestión de Base de Datos (SGBD); en inglés: Database Management System (DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos. Entre los SGBD disponibles en el mercado se encuentran: PostgreSQL, MySQL, Firebird, SQLite, Oracle, Microsoft SQL Server, entre otros.

#### **PostgreSQL.**

Es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD. Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que, mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto. Es un sistema objeto-relacional pues incluye aspectos del paradigma orientado a objetos, tales como la herencia, tipos de datos, funciones, restricciones, reglas e integridad transaccional, aunque no llega a ser un gestor con orientación a objetos pura. (8)

#### **MySQL**

Es un sistema de gestión de base de datos relacional, multiplataforma, multi-hilo y multi-usuario, con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. MySQL AB pertenece a Sun Microsystems desde enero del 2008. Por un lado lo ofrece GNU GPL, pero entidades que quieran incorporarlo en productos privados pueden comprar a la empresa una licencia que les permita ese uso. Está desarrollando en su mayor parte ANSIC (Estándar desarrollado por el Instituto Nacional Estadounidense de Estándares).

Al contrario de proyectos como Apache donde el software es desarrollado por una comunidad publica y el copyright del código está en el poder del autor individual.

MySQL está poseído y patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el licenciamiento anteriormente abordado. Además de las ventas de licencias privadas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius. (4)

### ***¿Por qué MySQL?***

MySQL cuenta con muchas ventajas, entre las que se destacan:

- Alto rendimiento: Es muy rápido con respecto a otros gestores de bases de datos.
- Bajo coste: Está disponible de manera gratuita, bajo una licencia de código abierto, o por un precio reducido en forma de licencia comercial si resultara necesario para su aplicación.
- Facilidad de configuración y aprendizaje: Las bases de datos más modernas utilizan SQL. Si ha utilizado otros RDBMS, no debería tener problemas para adaptarse a este sistema. MySQL resulta además más sencillo de configurar que otros productos similares.
- Portabilidad: Se puede utilizar en una gran cantidad de sistemas Unix diferentes así como bajo Microsoft Windows.

- Accesibilidad a código fuente: Como en el caso de PHP, puede obtener y modificar el código fuente de MySQL. (8)

### **1.2.6 Servidores para aplicaciones web**

El servidor web es un programa que corre sobre el servidor que escucha las peticiones HTTP (Hyper Text Markup Language) que le llegan y las satisface. Dependiendo del tipo de petición, el servidor web buscará una página web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición. Entre los servidores de aplicaciones web más usados se encuentran: Apache, Zope, IIS, y Tomcat. Se seleccionó Apache por las siguientes características:

#### **Apache**

Apache sustancialmente, es un proyecto nacido para crear un servidor de web estable, fiable y veloz para plataformas Unix. Apache nace, por una parte, de un código ya existente y de una serie de parches (patch) para mejorar su fiabilidad y sus características; de ahí su nombre.

#### Características

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache.
- Actualmente existen muchos módulos para Apache que son adaptables y están ahí para que se instalen cuando se necesiten. Otra cosa importante es que cualquiera que posea alguna experiencia en la programación de C o Perl (Practical Extracting and Reporting Language) puede escribir un módulo para realizar una función determinada.
- Trabaja con Perl, PHP y otros lenguajes de Script. Perl se destaca en el mundo del Script y Apache utiliza su parte del pastel de Perl tanto con soporte CGI (Commun Gateway Interfaces), como con Java y páginas JSP

(Paginas de Servidor Java). Teniendo todo el soporte que se necesita para tener páginas dinámicas.

- Permite personalizar la respuesta ante los errores que se puedan dar en el servidor. Es posible configurarlo para que se ejecute un determinado script cuando ocurra un error en concreto. (7)

Apache tiene amplia aceptación en la red: desde 1996, es el servidor HTTP más usado en el mundo y alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo. (7)

### **1.2.7 Paquete de tecnología XAMPP**

Es un paquete formado por un servidor web Apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Perl. El nombre proviene de X (para cualquier sistema operativo), A (Apache), M (MySQL), P (PHP), P (Perl). El programa esta liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X.

XAMPP es regularmente actualizado para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL, y PhpMyAdmin. Para instalar XAMPP requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar programas por separado.

Oficialmente, los diseñadores de XAMPP solo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios web y programadores testear su trabajo en sus propios ordenadores sin ningún acceso a Internet. En la práctica sin embargo, XAMPP es utilizado actualmente para servidor de sitios web en WWW, y con algunas modificaciones es generalmente lo suficientemente seguro para serlo. Una herramienta especial es suministrada para proteger fácilmente las partes más importantes del paquete. (5)

### **1.3 Metodologías de desarrollo de software**

Una metodología para el desarrollo de un proceso de software es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas informáticos. Las metodologías existentes en la actualidad se dividen en dos grandes grupos atendiendo a sus características: las metodologías tradicionales (RUP, MSF) y las metodologías ágiles (XP, SCRUM). Las primeras están pensadas para el uso exhaustivo de documentación durante todo el ciclo del proyecto mientras que las segundas ponen vital importancia en la capacidad de respuesta a los cambios, la confianza en las habilidades del equipo y al mantener una buena relación con el cliente (9)

#### **1.3.1 Metodología propuesta para el desarrollo de la aplicación.**

El desarrollo de software no es una tarea fácil, prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y anotaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en muchos otros.

Una posible mejora es incluir en los procesos de desarrollo más actividades, artefactos y restricciones, basando en los puntos débiles detectados, sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuáles dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas.

Este enfoque ha mostrado su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, no obstante, mantiene



una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales.

### **1.3.2 Proceso Unificado de Desarrollo (RUP)**

RUP es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML y trabajo de muchas metodologías utilizadas por los clientes.

El ciclo de vida de RUP se caracteriza por:

- Dirigido por Casos de Uso (CU)
- Centrado en la arquitectura
- Iterativo e incremental

Es práctico dividir el trabajo en partes más pequeñas o mini proyectos, cada uno de ellos es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada es por eso que se dice que son mini proyectos.

(4)

Ventajas de RUP:

- Es sencillo, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.
- Seguimiento detallado en cada una de las fases.
- No se necesita interacción con el cliente en todo el ciclo de desarrollo.
- Gran número de documentación que proporciona un control estricto de todo lo que pasó a lo largo del proyecto.
- Se define el alcance del proyecto desde un principio.

Desventajas de RUP:

- La evaluación de riesgos es compleja.
- El cliente deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto con él.

### **1.3.3 Programación Extrema (Extreme Programming, XP)**

XP es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, que describe la filosofía de XP en el "Manifiesto Ágil", sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. (10)

#### **1.3.3.1 Valores que promueve XP**

**Simplicidad:** XP propone el principio de hacer las cosas más simple que pueda funcionar, en relación al proceso y la codificación. Es mejor hacer hoy algo simple, que hacerlo complicado y probablemente nunca usarlo mañana.

**Comunicación:** Algunos problemas en los proyectos tienen su origen en que alguien no dijo algo importante en algún momento. XP hace imposible la falta de comunicación.

**Retroalimentación:** Retroalimentación concreta y frecuente del cliente, del equipo y de los usuarios finales da una mayor oportunidad de dirigir el esfuerzo eficientemente.

**Coraje:** El coraje (valor) existe en el contexto de los otros 3 valores.

#### **1.3.3.2 El ciclo de vida ideal de XP**

El ciclo de vida ideal de XP consta de seis fases:

- Exploración.
- Planificación de la Entrega (Reléase)
- Iteraciones.
- Producción.
- Mantenimiento.
- Muerte del Proyecto.

### **1.3.3.3 Prácticas en las que se fundamenta XP**

#### ***Planificación incremental***

La programación extrema asume que la planificación nunca será perfecta, y que variará en función de cómo varíen las necesidades del negocio. Por tanto, el valor real reside en obtener rápidamente un plan inicial, y contar con mecanismos de retroalimentación que permitan conocer con precisión dónde se está. Como es lógico, la planificación es iterativa: un representante del negocio decide al inicio de cada iteración qué características concretas se van a implementar.

El objetivo de XP es generar versiones de la aplicación tan pequeñas como sea posible, pero que proporcionen un valor adicional claro, desde el punto de vista del negocio. A estas versiones se les denomina *releases*.

Un *release* cuenta con un cierto número de historias. La historia es la unidad de funcionalidad en un proyecto XP, y corresponde a la mínima funcionalidad posible que tiene valor desde el punto de vista del negocio. Durante cada iteración se cierran varias historias, lo que hace que toda iteración agregue un valor tangible para el cliente.

Gran parte de la eficacia de este modelo de planificación se deriva de una división clara de responsabilidades, que tiene en cuenta las necesidades del negocio en todo momento. Dentro de esta división, el representante del cliente tiene las siguientes responsabilidades:

- Decidir qué se implementa en cada release o iteración.
- Fijar las fechas de fin de un release, recortando unas características o añadiendo otras.

- Priorizar el orden de implementación, en función del valor de negocio.

Las responsabilidades del equipo de desarrollo son las siguientes:

- Estimar cuánto tiempo llevará una historia de usuario, esto es fundamental para el cliente, y puede llevarle a reconsiderar qué historias se deben incluir en una iteración.
- Proporcionar información sobre el coste de utilizar distintas opciones tecnológicas.
- Organizar el equipo.
- Estimar el riesgo de cada historia.
- Decidir el orden de desarrollo de historias dentro de la iteración.

### ***Testing***

La ejecución automatizada de *tests* es un elemento clave de la metodología XP. Existen tanto tests internos (o tests de unidad), para garantizar que el mismo es correcto, como tests de aceptación, para garantizar que el código hace lo que debe hacer. El cliente es el responsable de definir los tests de aceptación, no necesariamente de implementarlos. Él es la persona mejor calificada para decidir cuál es la funcionalidad más valiosa.

Un efecto colateral importante de los tests, es que dan una gran seguridad a los desarrolladores; es posible llegar a hacer cambios más o menos importantes sin miedo a problemas inesperados, dado que proporciona una red de seguridad. La existencia de tests hace al código muy maleable.

### ***Diseño simple***

Una práctica fundamental de la programación extrema es utilizar diseños tan simples como sea posible. El principio es "utilizar el diseño más sencillo que consiga que todo funcione". La metodología XP nos pide, que no se viva bajo la ilusión de que un diseño puede resolver todas o gran parte de las situaciones futuras.

XP define un "diseño tan simple como sea posible" aquel que pasa todos los tests, no contiene código duplicado, deja clara la intención de los programadores (enfatisa el

qué, no el cómo) en cada línea de código y contiene el menor número posible de clases y métodos.

### ***Propiedad colectiva del código***

XP aboga por la propiedad colectiva del código. En otras palabras, todo el mundo tiene autoridad para hacer cambios a cualquier código, y es responsable de ellos. Esto permite no tener que estar esperando a otros cuando todo lo que hace falta es algún pequeño cambio.

### ***Integración continua***

En muchos casos la integración de código produce efectos colaterales imprevistos, y en ocasiones esta puede llegar a ser realmente traumática, cuando dejan de funcionar cosas por motivos desconocidos. La programación extrema hace que la integración sea permanente, con lo que todos los problemas se manifiestan de forma inmediata.

### ***Clientes en el equipo***

Algunos de los problemas más graves en el desarrollo son los que se originan cuando el equipo toma decisiones de negocio críticas. Esto no debería ocurrir, pero en el momento cumbre, con frecuencia no se obtiene retroalimentación del cliente con la fluidez necesaria.

La metodología XP intenta resolver este tipo de problemas integrando un representante del negocio dentro del equipo de desarrollo. Esta persona siempre está disponible para resolver dudas y para decidir qué se hace en cada momento, en función de los intereses del negocio.

### ***Entregas pequeñas***

Siguiendo la política de la metodología XP, de dar el máximo valor posible en cada momento, se intenta liberar nuevas versiones de las aplicaciones con frecuencia. Estas deben ser tan pequeñas como sea posible, aunque deben añadir suficiente valor, para que resulten meritorias para el cliente.

***Semana de 40 horas***

La programación extrema lleva un modo de trabajo en el que el equipo siempre está al 100%. Una semana de 40 horas en las que se dedica la mayor parte del tiempo a tareas que suponen un avance puede dar mucho de sí, y hace innecesario recurrir a sobreesfuerzos, excepto en casos extremos. Además, el esfuerzo continuado pronto lleva a un rendimiento menor y a un deterioro de la moral de todo el equipo.

Sirve de presentación del trabajo y orienta al lector sobre los aspectos fundamentales de este. Se explica con absoluta claridad y de forma corrida. Debe contener los siguientes aspectos (no necesariamente tienen que aparecer todos los puntos):

- Descripción de la organización donde se realiza el proyecto. (Caracterización de la organización, objeto social, misión, etc.)
- Situación problemática y problema a resolver.
- Actualidad y necesidad del trabajo.
- Antecedentes (deben indicarse las referencias bibliográficas utilizadas como fuente de información).
- Aportes prácticos esperados del trabajo.
- Objeto de estudio.
- Objetivos del trabajo (objetivo general y objetivos específicos).
- Tareas principales desarrolladas para cumplir los objetivos.
- Estudio de herramientas, lenguajes y tecnologías a utilizar.
  - Opciones existentes en la actualidad.
  - Selección para el desarrollo del trabajo y justificación de la misma.
- Efecto económico: exposición de los beneficios del Proyecto elaborado, los cuales pueden ser:
  - Beneficios de carácter técnico.
  - Beneficios intangibles (económicos, sociales o políticos).
  - Beneficios económicos tangibles. (Esto último es obligatorio para todos los trabajos, debiéndose especificar los beneficios o aportes económicos del trabajo).

#### **1.3.3.4 Fases de la Metodología XP**

##### **Fase I: Planificación**

- Se escriben historias de usuario, cuya idea principal es describir un caso de uso en dos o tres líneas con terminología del cliente (de hecho, se supone que deben ser escritos por el mismo), de tal manera que se creen *test* de aceptación para historias de usuarios (*user story*) y permita hacer una estimación de tiempo de desarrollo del mismo.
- Se crea un plan de lanzamiento (*release planning*), que debe servir para crear un calendario que todos puedan cumplir y en cuyo desarrollo hayan participado todas las personas involucradas en el proyecto. Se usa como base las historias de usuario, participando el cliente en la elección de las que se desarrollarán, y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto.
- El desarrollo se divide en iteraciones, cada una de las cuales comienzan con un plan de iteración, para el que se eligen las historias de usuario a desarrollar y las tareas de desarrollo.
- Se cambia el proceso cuanto sea necesario, para adaptarlo al proyecto.

##### **Fase II: Diseño**

- Se eligen los diseños funcionales más simples.
- Se elige una metáfora del sistema para que el nombrado de clases, siga una misma línea, facilitando la reutilización y la comprensión del código.
- Se escriben tarjetas de clase-responsabilidades-colaboración (CRC) para cada objeto, que permitan abstraerse al pensamiento estructurado y que el equipo de desarrollo completo participe en el diseño.

##### **Fase III: Codificación**

- El cliente está siempre disponible, de ser posible, cara a cara. La idea es que forme parte del equipo de desarrollo, y esté presente en todas las fases de XP. La idea es usar el tiempo del cliente para estas tareas en lugar de crear una

detallada especificación de requisitos, y evitar la entrega de un producto insuficiente, que le hará perder tiempo.

- El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilitando la comprensión y refactorización del código.
- Las pruebas unitarias se codifican antes que el código en sí, haciendo que la codificación de este último sea más rápida, y que cuando se afronte la misma se tenga más claro qué objetivos tiene que cumplir lo que se va a codificar.
- La programación del código se realiza en parejas, para aumentar la calidad del mismo. En cada momento, solo habrá una pareja de programadores que integre código.
- Se integra código y se lanza dicha integración de manera frecuente, evitando divergencias en el desarrollo y permitiendo que todo el mundo trabaje con la última versión del desarrollo. De esta manera, se evitará pasar grandes períodos de tiempo integrando el código al final del desarrollo, ya que las incompatibilidades serán detectadas enseguida.
- Se usa la propiedad colectiva del código, lo que se traduce en que cualquier programador puede cambiar cualquier parte del código. El objetivo es fomentar la contribución de ideas por parte de todo el equipo de desarrollo.
- Se deja la optimización para el final.
- No se hacen horas extra de trabajo.

### Fase IV: Pruebas

- Todo el código debe tener pruebas unitarias, y debe pasarlas antes de ser lanzado.
- Cuando se encuentra un error de codificación o bug, se desarrollan pruebas para evitar volver a caer en el mismo.
- Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las historias de usuarios (HU) elegidas para la iteración, y son "pruebas de caja negra", en las que el cliente verifica el correcto funcionamiento de lo que se está probando. Cuando se



pasa la prueba de aceptación, se considera que la correspondiente historia de usuario se ha completado. (11)

### **1.3.4 Fundamentación de la selección de la metodología de desarrollo de software.**

XP y RUP son dos grandes metodologías de desarrollo de software, pero después de analizar sus principales características y los aspectos más sobresalientes de cada una de ellas, se ha determinado la utilización de XP, una metodología ligera, con menos requerimientos de documentación y planificación para el desarrollo de la aplicación, pues RUP es más aplicable a proyectos de mayor envergadura. De escogerse RUP traería dificultades como son:

- Multitud de artefactos: El hecho de realizar varios artefactos y mantenerlos actualizados consume mucho tiempo.
- El poco personal de desarrollo: Al ser solo una persona a cargo del desarrollo de la aplicación, este tomaría varios roles en cada etapa y sería muy complejo cumplir con las actividades de cada uno de ellos.
- Se necesita: La documentación mínima necesaria para el futuro soporte y mantenimiento del producto final.
- Requisitos cambiantes: Los cambios en un proceso de desarrollo son inevitables, al aparecer un nuevo requisito hace que se tenga que comenzar una nueva iteración para dar cumplimiento a su funcionalidad. Como el proyecto está en plena investigación y en un ambiente de desarrollo sujeto a cambios repentinos se sugieren una gran adaptabilidad y pronta respuesta, lo cual RUP no ofrece.
- Planificación inexistente: La planificación que se realiza en las fases iniciales está sujeta a muchas variaciones en dependencia de los cambios que se experimenten en los requisitos. Por tanto se hace muy difícil planificar actividades específicas si no se tiene claro que se debe hacer realmente.

Los inconvenientes planteados pueden ser eliminados con la utilización de la metodología XP.

## **1.4 Framework de desarrollo web**

En el desarrollo de Software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Un framework web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo clases en Java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. (12)

### **1.4.1 Patrón MVC**

Para comprender como trabajan los frameworks web existentes es imprescindible conocer el patrón Modelo Vista Controlador (MVC).

MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la

página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.



**Figura 1 Modelo Vista Controlador**

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema. (13)

La mayoría de los frameworks para web implementan este patrón. Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.

3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta dirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. Este uso del patrón Observador no es posible en las aplicaciones web puesto que las clases de la vista están desconectadas del modelo y del controlador. En general el controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista. Por ejemplo en el MVC usado por Apple en su framework Cocoa. Suele citarse como Modelo-Interface-Control, una variación del MVC más puro.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente. (5)

#### **1.4.2 Tipos de Framework WEB**

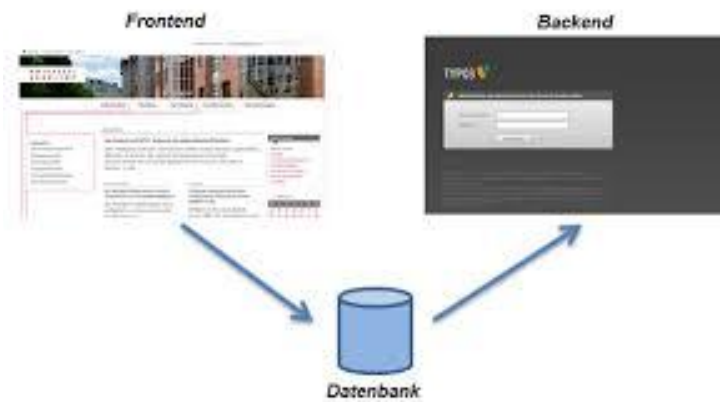
Existen varios tipos de frameworks web: orientados a la interfaz de usuario, como Java Server Faces, orientados a aplicaciones de publicación de documentos, como Coocoon, orientados a la parte de control de eventos, como Struts y algunos que incluyen varios elementos como Tapestry.

La mayoría de frameworks web se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC o con el modelo 2 de Servlets y JSP, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación. (13)

Básicamente, existen 2 tipos de diferenciar: backend y frontend (esta distinción se hace en función de si el framework es para la parte visual o para la parte lógica de la aplicación).

Es importante entender que los frameworks son un aspecto conceptual: un kit estándar prefabricado desde el que trabajar. El concepto de un framework se puede aplicar a diferentes procesos que se llevan a cabo en la web:

- Capa del backend developer que conecta la base de datos con el contenido del sitio y utiliza el lenguaje PHP.
- Capa del frontend developer que trata la parte visual de cara al usuario formada con los lenguajes HTML y CSS. (14)



**Figura 2 Frontend - Backend**

Así pues, un resumen sobre los tipos de Frameworks sería el siguiente:

### **Capa visual / Frontend**

Interfaces de usuario

Lenguajes: CSS, HTML, Javascript...

Frameworks: Bootstrap, Foundation3, Grid Systems, etc.

### **Capa lógica / Backend**

Operativa y lógica de la aplicación

**Lenguajes:** PHP, JAVA, PYTHON...

**Frameworks:** Symfony, Django, Yii, etc.

#### **1.4.2.1 Capa de datos**

Bases de datos, persistencia de datos.

Tecnologías: MySQL, PostgreSQL, NoSQL, etc. (14)

### **1.4.3 Características**

A continuación enunciamos una serie de características que podemos encontrar en prácticamente todos los frameworks existentes.

|                                |   |
|--------------------------------|---|
| Abstracción de URLs y sesiones | No es necesario manipular directamente las URLs ni las sesiones, el framework ya se encarga de hacerlo.   |
| Acceso a datos                 | Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en XML, etc.  |
| Controladores                  | La mayoría de frameworks implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables |

a las necesidades de un proyecto concreto.

Autenticación y control

Incluyen mecanismos para la identificación de usuarios mediante login y password y permiten restringir el acceso a determinadas páginas a determinados usuarios.

Internacionalización

Separación entre diseño y contenido

#### **1.4.4 CodeIgniter**

CodeIgniter es un conjunto de herramientas para personas que construyen su aplicación web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría si lo escribiese desde cero, proveyéndole un rico juego de librerías para tareas comúnmente necesarias, así como una interface simple y estructura lógica para acceder a esas librerías. CodeIgniter le permite creativamente enfocarse en su proyecto minimizando la cantidad de código necesaria para una tarea dada. Se encuentra bajo una licencia open source Apache/BSD-style (15)

Usa el acercamiento Modelo Vista Controlador, que permite una buena separación entre lógica y presentación. Esto es particularmente bueno para proyectos en los cuales los diseñadores están trabajando con sus archivos de plantilla, ya que el código en esos archivos será mínimo.

Verdaderamente liviano y rápido. El núcleo del sistema solo requiere unas pocas pequeñas librerías. Esto es en duro contraste a muchos entornos de trabajo que requieren significativamente más recursos. Las librerías adicionales son cargadas dinámicamente a pedido, basado en sus necesidades para un proceso dado, así que el sistema base es muy delgado y bastante rápido.

##### **1.4.4.1 Características**

- Sistema Basado en Modelo-Vista-Controlador

- Compatible con PHP 4
- Extremadamente Liviano
- Clases de base de datos llenas de características con soporte para varias plataformas.
- Soporte de Active Record para Base de Datos
- Formulario y Validación de Datos
- Manejo de Sesión
- Clase de Carga (upload) de Archivo
- Paginación
- Encriptación de Datos
- Historial de Errores
- Clase de Calendario
- Clase de Agente del Usuario
- Clase de Motor de Plantillas
- Clase de Prueba de Unidad
- URLs amigables a motores de búsqueda
- Ruteo de URI Flexible
- Larga librería de funciones "asistentes"

### 1.4.5 Zend Framework

Zend Framework es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado.

ZF ofrece un gran rendimiento y una robusta implementación MVC, una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores



puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. (16)

#### **1.4.5.1 Estructura de los componentes**

La estructura de los componentes de Zend Framework es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado.

A menudo se refiere a este tipo de diseño como "use-at-will" (uso a voluntad). Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de Zend Framework conforman un potente y extensible framework de aplicaciones web al combinarse. Zend Framework ofrece un gran rendimiento y una robusta implementación Patrón\_Modelo\_Vista\_Controlador, una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos.

Otros componentes, como Zend\_Auth y Zend\_Acl, proveen autenticación de usuarios y autorización diferentes a las tiendas de certificados comunes.

También existen componentes que implementan bibliotecas de cliente para acceder de forma sencilla a los servicios web más populares. Cualesquiera que sean las necesidades de su solicitud, usted tiene todas las posibilidades de encontrar un componente de Zend Framework que se pueda utilizar para reducir drásticamente el tiempo de desarrollo, con una base completamente sólida. El principal patrocinador del proyecto Zend Framework es Zend Technologies, pero muchas empresas han contribuido con componentes o características importantes para el marco. Empresas como Google, Microsoft y Strikelron se han asociado con Zend para proporcionar interfaces de servicios web y otras tecnologías que desean poner a disposición de los desarrolladores de Zend Framework.

#### **1.4.5.2 Requisitos**

Zend Framework requiere un intérprete PHP 5 con un servidor web configurado para manejar scripts PHP correctamente. Algunas características requieren extensiones adicionales o características de servidor web; en muchos casos el framework puede ser utilizado sin ellos, aunque el rendimiento puede sufrir o las características auxiliares pueden no ser completamente funcionales. Un ejemplo de dicha dependencia es `mod_rewrite` en un entorno Apache, que puede ser utilizado para ejecutar "pretty URL " como " `http://www.example.com/user/edit`". Si `mod_rewrite` no está habilitado, Zend Framework puede ser configurado para apoyar las URL como " `http://www.example.com?controller=user&action=edit`". La pretty URL, puede ser utilizada para acortar las URL de representación textual o para la optimización de los motores de búsqueda (SEO), pero no afectan directamente a la funcionalidad de la aplicación.

#### **1.4.5.3 Versión de PHP**

Zend recomienda PHP 5.2.3 o superior por mejoras en la seguridad críticas y en el rendimiento, aunque Zend Framework requiere solo PHP 5.1.4 o posterior. Zend Framework tiene una extensa colección de unidades de prueba, que puede ejecutar utilizando PHPUnit 3.0 o posterior.

El Zend Framework posee un gran seguimiento entre la comunidad de desarrolladores y está enfocado a las aplicaciones tipo web 2.0. Debido a su seguimiento masivo, soporte extensivo y base de usuario activa, Zend es conocido como "The PHP Company Posee propiedades importantes que están construidas para el desarrollo a nivel corporativo y requiere un gran conocimiento de PHP.

#### **1.4.6 Yii Framework**

Yii es un framework para PHP de alto rendimiento basado en componentes web para desarrollar aplicaciones de gran escala. Permite una máxima reusabilidad en la programación web y puede acelerar significativamente el proceso de desarrollo.

Yii es un framework genérico para programar web que puede ser utilizado para desarrollar virtualmente cualquier tipo de aplicaciones web. Ya que es liviano y está equipado con las soluciones más sofisticadas, está especialmente diseñado para trabajar con aplicaciones web de tráfico alto, como portales, foros, CMS, comercios electrónicos, etc. (17)

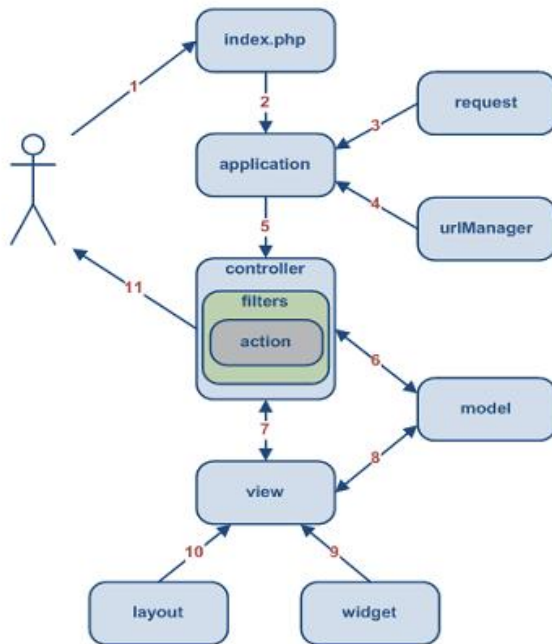
Como la mayoría de los framework para PHP, Yii es un framework que sigue el patrón de diseño MVC. Yii sobresale sobre los otros frameworks PHP por su eficiencia y su rica librería de funcionalidades así como también su clara documentación, puesto que es el resultado de la experiencia que tienen sus autores en el desarrollo de aplicaciones web ricas en funcionalidad, y la investigación y reflejo de los frameworks y aplicaciones más populares para programar aplicaciones web. Es un framework de desarrollo de aplicaciones web escrito en PHP usando el paradigma de programación POO (Programación Orientada a Objetos).

Se destaca frente a otros frameworks PHP por su **eficiencia, gran cantidad de características y baja curva de aprendizaje.**

Para crear aplicaciones utilizando Yii, debe seguir un simple proceso compuesto de 3 pasos:

1. Crear la base de datos
2. Generar el código PHP de la base, utilizando Yii.
3. Modificar el código para adecuarlo a sus necesidades.

#### 1.4.6.1 Funcionamiento del framework Yii:



**Figura 3** Funcionamiento de Yii Framework

1. Un usuario hace una petición en la siguiente URL: <http://www.example.com/index.php?r=post/show&id=1> Y el servidor web maneja la petición ejecutando el script index.php.
2. El script crea una instancia de la aplicación y la ejecuta.
3. La aplicación obtiene la información detallada de la petición de los usuarios desde un componente de la aplicación llamado request (petición).
4. La aplicación determina la petición del controlador y actúa con la ayuda de un componente de la aplicación llamado urlManager (manejador de url's).
5. Para este ejemplo, el controlador es Post que hace referencia a la clase PostController; y la acción es show (mostrar) del cuál su significado es determinado por el controlador.
6. La aplicación crea una instancia de la petición del controlador para más adelante manejar la petición del usuario. El controlador determina que la acción show (mostrar) se refiere a un método llamado actionShow en la clase del controlador.

7. Entonces crea y ejecuta filtros (por ej.: controles de acceso) asociados a esta acción. La acción es ejecutada si los filtros lo permiten.
8. La acción lee un modelo Post el cuál si ID es 1 en la base de datos.
9. La acción asocia una vista llamada show (mostrar) con el modelo Post.
10. La vista lee y muestra los atributos del modelo Post.
11. La vista ejecuta algunos artilugios.
12. Los resultados de la vista son embebidos en un diseño(layout)
13. La acción completa la prestación de la vista y muestra el resultado al usuario.

#### **1.4.6.2 Licencia**

Yii es liberado bajo la Nueva Licencia BSD (Cláusula 3 de la licencia). Esto significa que es posible utilizar de forma gratuita para desarrollar cualquier aplicación web de código abierto o [software privativo | software privativo.] El texto que figura en la documentación oficial de Yii tiene licencia para el público bajo la GNU Licencia Libre. En general, el contenido de la documentación Yii puede ser copiado, modificado y redistribuido siempre y cuando la nueva versión de subvenciones de las mismas libertades a los demás y reconozca a los autores del artículo de la documentación utilizada Yii.

#### **1.4.6.3 Principales Características**

- Usa el patrón de Arquitectura de Software MVC (Modelo Vista Controlador) que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.
- Provee un Nivel de Abstracción hacia la Base de Datos (DAO, Data Access Object) y se apoya en el patrón de arquitectura de datos Active Record.
- Incluye soporte para autenticación de usuarios basado en el modelo RBAC role-based access control o control de acceso basado en roles.
- Herramientas para automatización y validación de Formularios
- Integración con jQuery.

- Avanzado control de Themes
- Web services, herramientas para simplificar la tarea de generar y consumir servicios web en diferentes formatos.
- Internacionalización y Localización.
- Sistema de manejo de cache.
- Manejo de Errores.
- Manejo de Seguridad (Sql injection, XSS, CSRF, cookie tampering, etc.).
- Generador de código automático, muy útil a la hora de implementar formularios o bien operaciones CRUD. Además de que el código generado cumple XHTML standards.
- Amigable y extensible, por lo cual se pueden usar sin problemas librerías de terceros como Pear, Zend, etc.
- El manejo de errores y logging. Los errores son manejados y personalizados, y los log de mensajes pueden ser categorizados, filtrados y movidos a diferentes destinos.
- Las medidas de seguridad incluyen la prevención cross-site scripting (XSS), prevención cross-site request forgery (CSRF), prevención de la manipulación de cookies, etc.
- Herramientas para pruebas unitarias y funcionales basados en PHPUnit y Selenium.
- Generación automática de código para el esqueleto de la aplicación, aplicaciones CRUD, etc.
- Generación de código por componentes de Yii y la herramienta por línea de comandos cumple con los estándares de XHTML.
- Cuidadosamente diseñado para trabajar bien con código de terceros. Por ejemplo, es posible usar el código de PHP o Zend Framework en una aplicación Yii. (17)

#### **1.4.6.4 Requerimientos**

Para correr una aplicación web Yii, usted necesita tener un servidor web con soporte PHP 5.1.0 o superior.

Para desarrolladores que deseen utilizar Yii, el entendimiento de Programación Orientada a Objetos (OOP) será de gran ayuda ya que Yii es un framework totalmente basado en OOP. (18)

#### **1.4.7 Fundamentación de la selección del Framework Yii**

Después de analizados los distintos frameworks de desarrollo web, sus características y requisitos, optamos por seleccionar el framework Yii. Ya que este es un framework PHP basado en componentes de alta performance para desarrollar aplicaciones web.

El mismo permite la máxima reutilización en la programación web y puede acelerar el proceso de desarrollo. Es un framework open source de programación utilizado para desarrollar todo tipo de aplicaciones web. El nombre Yii (acrónimo de Yes it is!) (Pronunciado /i:/) es por fácil (en inglés: *easy*), eficiente (en inglés: *efficient*) y extensible (en inglés: *extensible*). (18)

##### **1.4.7.1 Para qué es bueno utilizar Yii**

Yii es un framework genérico de programación web que puede ser utilizado para todo tipo de aplicaciones web. Gracias a que es liviano de correr y está equipado con soluciones de cacheo sofisticadas, (18) es adecuado para desarrollar aplicaciones de todo tipo.

##### **1.4.7.2 ¿Cómo se compara Yii con otros frameworks?**

Como la mayoría de los frameworks PHP, Yii es un framework MVC (modelo-vista-controlador).

Yii sobresale frente a frameworks PHP en su eficiencia, su gran cantidad de características y su clara documentación. Yii ha sido diseñado cuidadosamente desde el principio para el desarrollo de aplicaciones de web. No es ni un subproducto de un

proyecto ni un conglomerado de trabajo de terceros. Es el resultado de la vasta experiencia de los autores en desarrollo de aplicaciones web y de la investigación y la reflexión de los más populares frameworks de programación web y aplicaciones.

Debido a sus características, su alto rendimiento y su facilidad de uso y aprendizaje, se decidió que la aplicación se desarrollará utilizando el framework de desarrollo Yii Framework.

#### **1.4.8 Herramientas CASE**

Embarcadero ER-Studio: Es una herramienta de modelado de datos, se usa para el diseño y la construcción lógica y física de bases de datos. Su ambiente es de gran alcance y multinivel. Simple y fácil al usuario, ayuda a las organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia y alcanza en última instancia usos de más alta calidad que entreguen datos más eficientes y exactos a la empresa. (19)

Ventajas:

- Si se está comenzando un nuevo diseño o está manteniendo una base de datos existente, ER/Studio se combina con las características para ayudarle a conseguir que el trabajo sea hecho con eficacia.
- La creación de diagramas es clara y rápida.
- Tiene la posibilidad de realizar diagramas con desempeño rápido. (20)

### **1.5 Herramientas a utilizar en la propuesta de solución**

Luego de realizado un estudio detallado sobre las principales herramientas y metodologías de desarrollo de software, según las características más adecuadas para una satisfactoria fabricación del sistema que se desea obtener, se determinó que el desarrollo del proyecto estará guiado por la metodología XP; para el modelado de datos se utilizará la herramienta CASE Embarcadero ER/Studio; el lenguaje de programación será PHP, y HTML para generar las vistas, con el conjunto de estilos Metro-Bootstrap,



utilizando el framework de desarrollo Yii; además, el IDE de programación NetBeans 7.3; y el paquete XAMPP, donde se encuentra el servidor web Apache y el servidor de base de datos MySQL.

### ***Conclusiones del Capítulo***

En este capítulo se ha hecho referencia a los procesos del manejo de la información generada en la VRIP sobre la actividad científica y la innovación tecnológica, se detallaron las condiciones y problemas actuales que rodean el objeto de estudio, además se reflejaron los sistemas desarrollados vinculados a la investigación, y por último se han fundamentado las herramientas que se utilizarán para el diseño y desarrollo del producto.

## **CAPÍTULO II – PLANEACIÓN Y DISEÑO**

En este capítulo, se presenta la fase de planeación y diseño, donde se detallan las necesidades del cliente, se describen las funcionalidades que serán objeto de automatización mediante el empleo de las historias de usuarios (HU), se realiza una estimación del esfuerzo necesario para las mismas y se establece un plan de iteraciones necesarias sobre el sistema, para su terminación.

### **2.1 Funcionalidades generales**

La gestión de la información referente a la actividad científica y la innovación tecnológica que se realiza en la VRIP se realiza de forma manual, guardando las informaciones en planillas en papel o documentos digitales aislados, lo cual hace que el proceso se vuelva lento y difícil para el especialista encargado de realizarlo, esto, además, trae consigo que las consultas no sean eficientes, en algunos casos trae como consecuencia pérdida de información y deterioro de la misma. De la descripción anterior se identifican como funcionales generales del sistema guardar datos sobre los proyectos, artículos, eventos, entre otras; así como realizar búsquedas.

#### **2.1.1 Personal relacionado con el sistema**

Tabla 2.1 Personas relacionadas con el sistema

| <b>Personas relacionadas con el sistema</b> | <b>Justificación</b>   |
|---|--|
| Administrador                               | Es la persona que tiene permiso para realizar cualquier operación con el sistema.            |
| Secretaria                                  | Es la encargada de recopilar la información. Puede enviar, modificar y publicar información. |
| Desarrollador                               | Es la persona responsable de llevar a cabo la implementación del sistema.                    |

### **2.1.2 Lista de reserva**

Después de conocer el personal relacionado con el sistema, se procede a realizar el análisis de las funcionalidades que debe cumplir la aplicación para dar respuesta a los mismos. Para ello se enumerarán mediante una lista de reserva, las funcionalidades que el sistema debe ser capaz de cumplir.

De acuerdo a lo antes expuesto, el sistema debe ser capaz de:

1. Autenticar usuarios.
2. Insertar usuario.
3. Listar usuarios.
4. Modificar usuario.
5. Eliminar usuario.
6. Insertar línea de investigación.
7. Eliminar línea de investigación.
8. Modificar línea de investigación.
9. Listar línea de investigación.
10. Insertar indicadores dinámicos de evaluación.
11. Eliminar indicadores dinámicos de evaluación.
12. Modificar indicadores dinámicos de evaluación.
13. Listar indicadores dinámicos de evaluación.
14. Insertar proyecto.
15. Eliminar proyecto.
16. Modificar proyecto.
17. Listar proyectos.
18. Insertar secuencia de actividades.
19. Eliminar secuencia de actividades.
20. Modificar secuencia de actividades.
21. Listar secuencia de actividades.
22. Insertar miembro de proyecto.
23. Eliminar miembro de proyecto.

24. Modificar miembro de proyecto.
25. Listar miembros de proyecto.
26. Insertar documento.
27. Eliminar documento.
28. Modificar documento.
29. Listar documento.
30. Insertar artículo.
31. Eliminar artículo.
32. Modificar artículo.
33. Listar artículo.
34. Insertar premio.
35. Eliminar premio.
36. Modificar premio.
37. Listar premio.
38. Registrar software.
39. Listar registro de software.
40. Registrar publicación.
41. Listar registros de publicaciones.
42. Insertar evento.
43. Eliminar evento.
44. Modificar evento.
45. Listar evento.
46. Insertar plan de resultado.
47. Eliminar plan de resultado.
48. Modificar plan de resultado.
49. Listar plan de resultado.

### **2.1.3 Características del Sistema**

1. Requerimientos de software.
  - a. Navegador que soporte las características del lenguaje HTML5 y CSS3.
  - b. Servidores de bases de datos MySQL 5.x.

2. Requerimientos de apariencia e interfaz externa.
  - a. Diseño gráfico no debe ser complejo.
  - b. Utilizar colores que proporcionen una interfaz amigable al usuario.
3. Requerimientos de seguridad
  - a. Identificar al usuario antes de que pueda entrar al sistema
  - b. Garantizar que la información sea editada solo por quién tiene privilegios
  - c. Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel del usuario activo.
  - d. Verificaciones sobre acciones irreversibles como eliminaciones.
4. Requerimientos de usabilidad
  - a. Mostrar en cada página un título con el nombre de la sección correspondiente.
  - b. Resaltar en todos los formularios los campos obligatorios con un asterisco o con un texto en rojo.
  - c. Agregar etiquetas a los campos de entrada de datos con una breve descripción del formato que admiten y un ejemplo.
  - d. Verificar la información de todos los campos de entrada de datos, señalar aquellos con datos erróneos y mostrar mensajes que ayuden a corregirlos.
5. Rendimiento: Los tiempos de respuestas no son determinantes.
6. Portabilidad: Necesidad de que el sistema sea multiplataforma.

#### **2.1.4 Historias de usuario**

Las historias de usuarios (HU), son la técnica utilizada en XP para detallar los requisitos del software. Son el resultado directo del intercambio entre los usuarios y desarrolladores a través de reuniones donde las conocidas *tormenta de ideas (brain storm)* arrojan no solo los requerimientos, sino también las posibles soluciones; representan una forma rápida de administrar las necesidades de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para gestionarlos, debido a que un requerimiento de software es descrito de forma concreta y sencilla utilizando el lenguaje común del usuario. Las HU permiten responder ágilmente a los requerimientos cambiantes y aunque se redactan desde las

perspectivas de los clientes, también los desarrolladores pueden brindar ayuda en la identificación de las mismas.

**1. Historia de Usuario No.3:** Gestión de indicadores dinámicos de evaluación.

**Tabla 2.2 Historia de Usuario No.1**

|  |                              |
|--|------------------------------|
| Historia de usuario  |                              |
| Número: 3  | Usuario: administrador.      |
| Nombre: Gestión de indicadores dinámicos de evaluación.                        |                              |
| Prioridad en el negocio: Alta  | Riesgo en desarrollo: Alta.  |
| Puntos estimados: 2  | Iteración asignada: Primera. |
| Programador responsable: Deysi Columbié Hernández.                             |                              |
| Descripción: El usuario debe insertar los indicadores dinámicos de evaluación. |                              |
| Observaciones: Confirmado por el cliente.                                      |                              |

**2.2 Planificación de entregas**

En esta fase se establece la prioridad de cada HU, y a continuación, se realiza una estimación del esfuerzo necesario de cada una de ellas por parte de los programadores. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debe obtenerse en no más de dos a tres meses.

Las estimaciones asociadas a la implementación de las historias se establecen empleando como medida el punto de estimación. Un punto de estimación equivale a una semana ideal de programación, donde los miembros de los equipos de desarrollo, trabajan el tiempo planeado sin ningún tipo de interrupción, este punto de estimación

que se utiliza para representar la semana ideal, es de 5 días. Las historias generalmente tienen un valor de 1 a 3 puntos. Además, se mantiene un registro de la velocidad de desarrollo, establecida por puntos de iteración, basado fundamentalmente en la suma de los puntos de estimación correspondientes a las HU, que fueron terminadas en la última iteración.

### **2.2.1 Estimación de esfuerzo por historias de usuario**

Para el buen desarrollo del sistema propuesto, se realizó una estimación para cada una de las HU identificadas, y se obtienen los resultados que se muestran a continuación:

**Tabla 2.3 Esfuerzo por HU**

| <b>Historias de usuario</b>          | <b>Puntos de estimación</b> |
|--------------------------------------|-----------------------------|
| Autenticar usuarios                  | 1 día                       |
| Gestionar usuarios                   | 3 días                      |
| Gestionar línea de investigación     | 1 semana                    |
| Gestionar indicadores                | 1 semana                    |
| Gestionar proyectos de investigación | 1 semana                    |
| Gestionar secuencia de actividades   | 1 semana                    |
| Gestionar miembros de proyecto       | 1 semana                    |
| Gestionar documentos                 | 1 semana                    |
| Gestionar plan de resultado          | 1 semana                    |
| Gestionar artículo                   | 1 semana                    |
| Gestionar premio                     | 1 semana                    |
| Gestionar evento                     | 1 semana                    |
| Registrar software                   | 3 días                      |
| Registrar publicación                | 3 días                      |
| Listar registros de software         | 3 días                      |
| Listar registros publicaciones       | 3 días                      |



## **2.2.2 Planificación de iteraciones**

A partir de las HU antes expuestas y la estimación del esfuerzo propuesto para la realización de las mismas, se procede a realizar la planificación de la etapa de implementación del sistema, apoyándose en el tiempo e intentando concentrar las funcionalidades relacionadas en una misma iteración. En este plan se establece cuántas iteraciones serán necesarias realizar sobre el sistema para su terminación. El plan de iteraciones puede contener indicaciones sobre cuáles HU se incluirán en un *release*, lo cual debe ser consistente con el contenido de una o dos iteraciones.

En relación con lo antes tratado se decide realizar el sistema en 3 iteraciones, las cuales se explican de forma detalla a continuación:

### ***Primera iteración:***

Esta iteración tiene como objetivo darle cumplimiento a las HU que se consideraron de mayor importancia para el desarrollo de la aplicación. Al concluir dicha iteración se contará con todas las funcionalidades descritas en las HU 1, 2, 3, 4 y 5, las cuales hacen alusión a la inserción de la información de los usuarios, líneas de investigación, indicadores, y proyectos de investigación.

### ***Segunda iteración:***

Esta iteración tiene como finalidad desarrollar las HU 6, 7, 8, 10, 14, 15. Las mismas son las que brindan las funcionalidades de registrar la información de la secuencia de actividades, los documentos, el plan de resultado, los premios y los registros de software.

### ***Tercera iteración:***

Esta iteración tiene como finalidad desarrollar la HU 9, 11, 12, 13, 16 y 17. Las mismas van a tomar como referencia los miembros, los diferentes artículos, eventos, y registro de publicaciones.

**2.2.3 Plan de duración de las iteraciones**

Como parte del ciclo de vida de un proyecto guiado por la metodología de desarrollo de software XP, se crea el plan de duración de cada una de las iteraciones que se llevarán a cabo durante el desarrollo del mismo. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas las HU en cada una de las mismas.

Plan de duración de las iteraciones

**Tabla 2.4 Plan de duración de las iteraciones**

| <b>Iteración</b>   | <b>Historia de Usuario</b>         | <b>Duración total</b> |
|--------------------|------------------------------------|-----------------------|
| <b>Iteración 1</b> | Gestionar usuarios.                | 4 semanas y 3 días    |
|                    | Gestionar línea de investigación.  |                       |
|                    | Gestionar indicadores.             |                       |
|                    | Gestionar proyectos.               |                       |
|                    | Gestionar secuencia de actividades |                       |
| <b>Iteración 2</b> | Gestionar miembros de proyecto     | 4 semanas y 1 día     |
|                    | Gestionar documentos.              |                       |
|                    | Gestionar plan de resultados.      |                       |
|                    | Registrar software                 |                       |
|                    | Listar registros de software       |                       |
| <b>Iteración 3</b> | Gestionar artículo.                | 4 semanas y 1 días    |
|                    | Gestionar evento.                  |                       |
|                    | Gestionar premios                  |                       |

|  |                                   |  |
|--|-----------------------------------|--|
|  | Registrar publicación             |  |
|  | Listar registros de publicaciones |  |

### **2.3 Clases, responsabilidades y colaboración**

En este epígrafe tiene lugar la realización de las tarjetas de Clases, Responsabilidades y Colaboración, conocidas tradicionalmente como *tarjetas CRC*, las cuales se realizan con el objetivo de facilitar la comunicación y documentar los resultados. Además, las mismas permiten la total participación y contribución del equipo de desarrollo en la tarea de diseño. [\[Ver Anexo 2\]](#)

**Tabla 2.5 Tarjetas CRC Gestionar indicadores dinámicos de la evaluación.**

|  |               |
|--|---------------|
| Nombre de la Clase: Gestionar Indicadores dinámicos de la evaluación |               |
| Tipo de clase:   |               |
| Lógica del negocio   |               |
| Responsabilidades  | Colaboradores |
| Inserta los indicadores dinámicos de la evaluación                   | Insertar_ide  |
| Muestra indicadores dinámicos de la evaluación                       | Mostrar_ide   |
| Modifica los indicadores dinámicos de la evaluación                  | Modificar_ide |
| Eliminar indicadores dinámicos de la evaluación                      | Eliminar_ide  |

### ***Conclusiones del Capítulo***

Como resultado de este capítulo se abordó la fase de planeación y diseño donde se delinearon las historias de usuarios (HU) con la participación del cliente, se llevó a efecto la planificación de iteraciones de cada HU a partir de la estimación del esfuerzo necesario de las mismas, culminando así esta fase y se determina que el equipo de trabajo está listo para pasar a la siguiente etapa de desarrollo.

## **CAPÍTULO III – DESARROLLO Y PRUEBAS**

En este capítulo se inicia la fase de desarrollo y pruebas conforme a la metodología XP. Se presenta el modelo de datos empleado para la aplicación concluyente, se presentan además las tarjetas, clases, responsabilidades y colaboradores, que permitirán trabajar con una metodología basada en objetos, se realiza el desarrollo de las iteraciones a partir del desglose de las HU en tareas. Se muestran las interfaces gráficas de usuario diseñadas para la aplicación final. Se describen igualmente las pruebas realizadas y se indican las respuestas de la aplicación en el empleo de las diferentes funcionalidades, así como los posibles mensajes de error, información o aceptación que emite la misma cuando se utiliza una de estas funcionalidades.

### **3.1 Modelo de datos**

A continuación mostramos el modelo de datos utilizado en el sistema. En la gestión de información relacionada con la actividad científica e innovación tecnológica en la VRIP. Es utilizada la tabla **usuarios** para almacenar los datos de los usuarios que tendrán acceso al sistema. También se tiene la tabla **proyecto\_investigacion** que almacena la información de los proyectos que se presentan en la VRIP, **linea\_inv** que tiene los datos sobre los cuales se realiza la el proyecto. La tabla **secuencia\_actividades**, la cual guarda las actividades que se deben realizar en el proyecto y **miembro\_proyecto** la cual tiene la información de los miembros del proyecto. Los indicadores del proyecto se guardan en la tabla **indicador**, el cual tiene un plan cuyos datos van a la tabla **plan\_indicadores**. Los resultados se guardan en la tabla **plan\_resultado**.

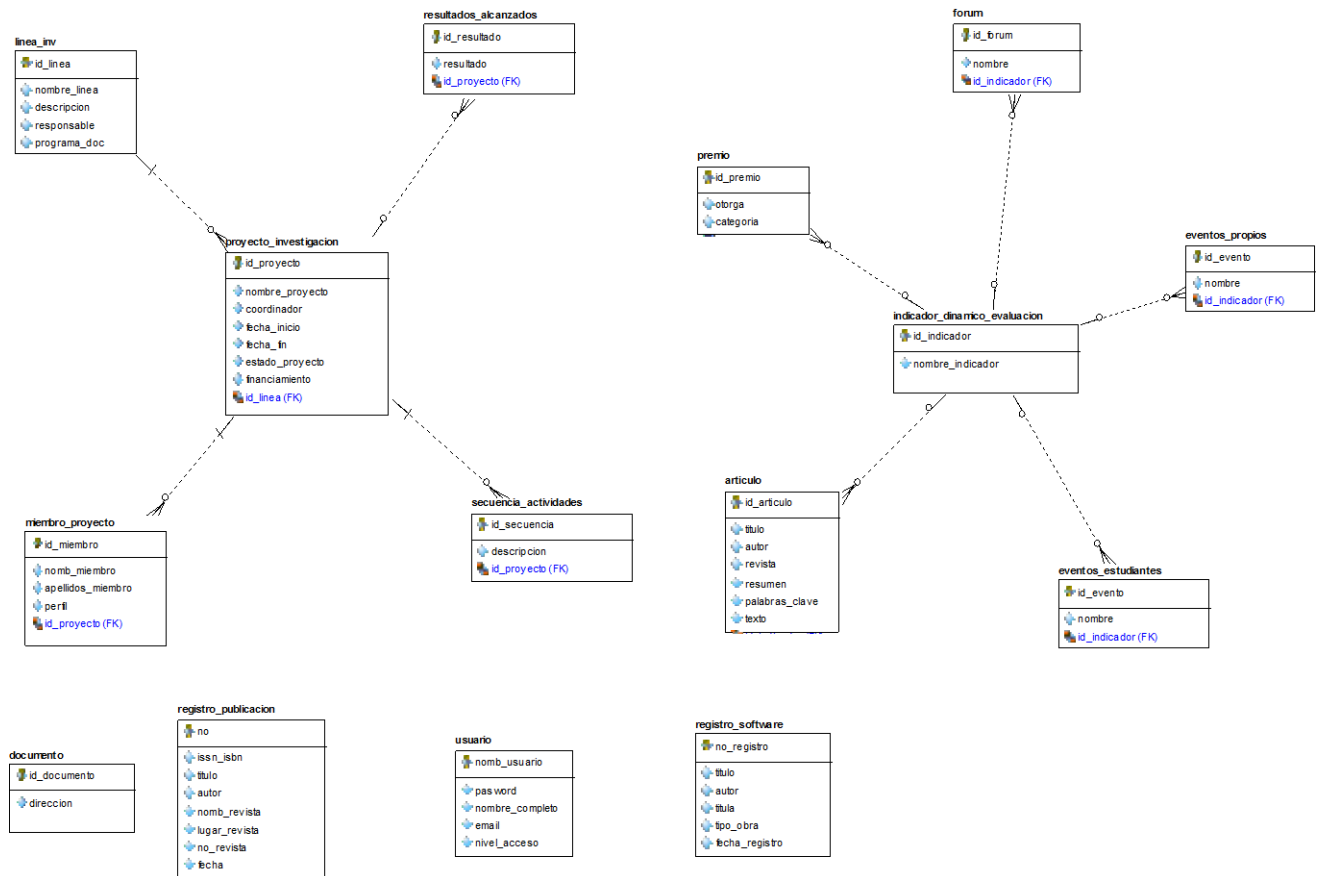


Figura 4 Modelo de datos

### 3.3 Desarrollo de las iteraciones

Durante la fase planificación y diseño fueron detalladas las HU correspondientes a cada una de las iteraciones a desarrollar, teniendo en cuenta las prioridades y restricciones de tiempo, previstas por el cliente.

### 3.4 Tareas por historias de usuario

Dentro del contenido de este plan, las HU se descomponen en tareas de programación o ingeniería, y a su vez estas son asignadas al equipo de desarrollo para su implementación. Las tareas no tienen que ser entendidas necesariamente por el cliente, pues las mismas, solo son utilizadas por los miembros del equipo de desarrollo, por lo que pueden ser escritas en lenguaje técnico. Estas se representan mediante las tarjetas de tareas.

**Tabla 3.1 Distribución de tareas de ingeniería por historia de usuario**

| <b>Historias de usuario</b>                 | <b>Tareas</b>   |
|---|---|
| <b>Gestionar usuarios</b>                   | <ul style="list-style-type: none"><li>• Insertar usuario.</li><li>• Modificar usuario.</li><li>• Eliminar usuario.</li><li>• Mostrar usuarios registrados.</li><li>• Cambiar contraseña.</li></ul>                  |
| <b>Gestionar línea de investigación</b>     | <ul style="list-style-type: none"><li>• Insertar línea de investigación.</li><li>• Modificar línea de investigación.</li><li>• Eliminar línea de investigación.</li><li>• Mostrar línea de investigación.</li></ul> |
| <b>Gestionar indicadores</b>                | <ul style="list-style-type: none"><li>• Insertar indicador.</li><li>• Modificar indicador.</li><li>• Eliminar indicador.</li><li>• Mostrar indicador.</li></ul>   |
| <b>Gestionar proyectos de investigación</b> | <ul style="list-style-type: none"><li>• Insertar proyecto.</li><li>• Modificar proyecto.</li><li>• Eliminar proyecto.</li><li>• Mostrar proyectos.</li></ul>  |
| <b>Gestionar secuencia de actividades</b>   | <ul style="list-style-type: none"><li>• Insertar actividades.</li><li>• Modificar actividades.</li><li>• Eliminar actividades.</li><li>• Mostrar actividades.</li></ul>   |
| <b>Gestionar miembros de proyecto</b>       | <ul style="list-style-type: none"><li>• Insertar miembros.</li><li>• Modificar miembros.</li><li>• Eliminar miembros.</li><li>• Mostrar miembros.</li></ul>   |
| <b>Gestionar documentos</b>                 | <ul style="list-style-type: none"><li>• Insertar investigación.</li><li>• Modificar investigación.</li><li>• Eliminar investigación.</li><li>• Mostrar investigación.</li></ul>                                     |

|  |  |
|--|--|
| <b>Gestionar plan de resultado</b>       | <ul style="list-style-type: none"> <li>• Insertar tipo de evento.</li> <li>• Modificar tipo de evento.</li> <li>• Eliminar tipo de evento.</li> <li>• Mostrar tipo de evento.</li> </ul> |
| <b>Gestionar artículo</b>                | <ul style="list-style-type: none"> <li>• Insertar artículo.</li> <li>• Modificar artículo.</li> <li>• Eliminar artículo.</li> <li>• Mostrar artículo.</li> </ul>                         |
| <b>Gestionar premio</b>                  | <ul style="list-style-type: none"> <li>• Insertar premio.</li> <li>• Modificar premio.</li> <li>• Eliminar premio.</li> <li>• Mostrar premio.</li> </ul>                                 |
| <b>Gestionar evento</b>                  | <ul style="list-style-type: none"> <li>• Insertar evento.</li> <li>• Modificar evento.</li> <li>• Eliminar evento.</li> <li>• Mostrar evento.</li> </ul>                                 |
| <b>Registrar software</b>                | <ul style="list-style-type: none"> <li>• Registrar software</li> </ul>   |
| <b>Listar registro de software</b>       | <ul style="list-style-type: none"> <li>• Listar registros de software</li> </ul>   |
| <b>Registrar publicaciones</b>           | <ul style="list-style-type: none"> <li>• Registrar publicaciones</li> </ul>  |
| <b>Listar registros de publicaciones</b> | <ul style="list-style-type: none"> <li>• Listar registros de publicaciones</li> </ul>  |

Tabla 3.2 Historias abordadas en la primera iteración

| <b>Historias de usuario</b>               | <b>Tiempo de estimación</b> |               |
|---|-----------------------------|---------------|
|   | <b>Estimación inicial</b>   | <b>Real</b>   |
| <b>Gestión de usuarios.</b>               | 3 días                      | <b>3 días</b> |
| <b>Gestión de línea de investigación.</b> | 5 días                      | <b>5 días</b> |
| <b>Gestión de indicadores.</b>            | 5 días                      | <b>5 días</b> |
| <b>Gestión de proyectos.</b>              | 5 días                      | <b>5 días</b> |
| <b>Gestionar secuencia de actividades</b> | <b>5 días</b>               | <b>5 días</b> |



Tabla 3.3 Historias de usuario abordadas en la segunda iteración

| Historias de usuario           | Tiempo de estimación |        |
|--------------------------------|----------------------|--------|
|                                | Estimación inicial   | Real   |
| Gestionar miembros de proyecto | 5 días               | 5 días |
| Gestionar documentos.          | 5 días               | 5 días |
| Gestionar plan de resultados.  | 5 días               | 5 días |
| Registrar software             | 5 días               | 5 días |
| Listar registros de software   | 5 días               | 5 días |

Tabla 3.4 Historias de usuario abordadas en la tercera iteración

| Historias de usuario              | Tiempo de estimación |        |
|-----------------------------------|----------------------|--------|
|                                   | Estimación inicial   | Real   |
| Gestionar artículo.               | 5 días               | 5 días |
| Gestionar evento.                 | 5 días               | 5 días |
| Gestionar premios                 | 5 días               | 5 días |
| Registrar publicación             | 3 días               | 3 días |
| Listar registros de publicaciones | 3 días               | 3 días |

### 3.5 Pruebas

En la Programación Extrema es esencial el desarrollo de las pruebas, permitiendo probar continuamente el código. Cada vez que se desea implementar las funcionalidades que tendrá el software, XP propone una redacción sencilla de prueba, para ser pasada por el código posteriormente. El proceso constante de las pruebas permite la obtención de un producto con mayor calidad, y se ofrece a los programadores una mayor certeza en el trabajo que desempeñan. En la metodología XP hay dos tipos de pruebas; las unitarias o desarrollo dirigido por pruebas, desarrolladas por los programadores verificando su código de forma automática, y las

pruebas de aceptación, las cuales son evaluadas luego de culminar una iteración, se verifica así, que se cumplió la funcionalidad requerida por el cliente. Con estas normas se obtiene un código simple y funcional de manera bastante rápida y eficiente. Por esto es importante pasar las pruebas al 100%.

### **3.5.1 Pruebas de aceptación**

Las pruebas de aceptación en XP, se pueden asociar con las pruebas de caja negra que se aplican en otras metodologías de desarrollo, solo que se crean a partir de las historias de usuario y no por un listado de requerimientos. Durante las iteraciones, las HU se traducen a pruebas de aceptación. En ellas se especifican desde la perspectiva del cliente, los escenarios para probar que la HU ha sido implementada correctamente. La misma puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo que persiguen estas pruebas, es garantizar que las funcionalidades solicitadas por el cliente han sido realizadas. Una HU no se considera completa hasta que no ha transitado por sus pruebas de aceptación. Luego de ver los arquetipos anteriores empleados para la realización de las pruebas y reunirse con el cliente para su análisis, el mismo decidió que se lleve a cabo el proceso mediante las pruebas de aceptación. La planilla utilizada para plasmar el contenido de las pruebas de aceptación se muestra a continuación.

**Tabla 3.5 Prueba de aceptación.**

|   |
|---|
| <b>Prueba de aceptación</b>   |
| HU: Gestionar proyecto  |
| Nombre: Prueba para comprobar la entrada de datos del proyecto.   |
| Descripción: Validación de entrada de los datos del proyecto.   |
| Condiciones de ejecución: El usuario debe introducir los datos del proyecto.  |
| Entrada/Pasos de ejecución: el usuario intenta entrar los datos.  |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si se insertan los datos correctamente el sistema limpia los campos para que el usuario entre otros datos nuevos.</li></ul> |

Se emite un mensaje de error en caso de que:

- Se dejen campos obligatorios vacíos.
- Se intente introducir un proyecto que ya exista.

Evaluación de la prueba: Aceptada.

### ***Conclusiones del Capítulo***

Con el desarrollo de este capítulo se llevó a cabo la fase de desarrollo y diseño donde se presenta el modelo de datos de la aplicación a obtener, logrando una visión detallada de sus atributos y las relaciones entre sus clases, además, las principales clases mediante el empleo de las tarjetas CRC. Se formaliza el desarrollo de las iteraciones a partir de la distribución de tareas de ingeniería por HU, y se les realiza las pruebas de aceptación a las funcionalidades de mayor importancia.

## CAPÍTULO IV - ESTUDIO DE FACTIBILIDAD

### 4.1 Introducción

Para el estudio de la factibilidad de este proyecto se utiliza la **Metodología Costo Efectividad (Beneficio)**, la cual permite establecer a través de la investigación unida de dos factores, el beneficio a la hora de la ejecución de un proyecto.

- **El costo:** Contiene la ejecución de la aplicación informática, su adquisición, la puesta en marcha del sistema hardware/software y los costos de operación.
- **La efectividad:** Es el contenido que posee el proyecto para satisfacer los diversos problemas o alcanzar los objetivos por lo cual se creó, es decir un proyecto será más o menos efectivo si se le dan mayor o menor cumplimiento al alcance, con la finalidad, para la cual fue ideado (costo por unidad de cumplimiento del objetivo). (4)

El desarrollo de un producto informático, siempre tiene un costo. Este puede estar justificado por los beneficios tanto tangibles como intangibles que origina el mismo. En este proceso, se necesita de una selección adecuada de los elementos más convenientes para su evaluación.

### 4.2 Efectos Económicos

- Efectos directos.
- Efectos indirectos.
- Efectos externos.
- Intangibles.

#### 4.2.1 Efectos directos

##### 4.2.1.1 Positivos

- ✓ Centralización de la información.
- ✓ Facilidad en la obtención de información.
- ✓ Rapidez en las consultas a los datos.
- ✓ Disminución del volumen de documentos impresos.

- ✓ Existencia de un control de usuario para acceder a las informaciones.

#### **4.2.1.2 Negativos**

- ✓ Para el uso de esta aplicación implementada en plataforma web se utilizan los navegadores preferiblemente Mozilla Firefox, porque se adapta a cualquier tipo de diseño.
- ✓ Para el uso de esta aplicación es imprescindible un ordenador conectado a la red, llevando esto consigo gastos en consumo de energía eléctrica.

#### **4.2.2 Efectos indirectos**

- ✓ Los efectos económicos observados que pudiera repercutir sobre otros mercados no son perceptibles, aunque este proyecto no está construido con la finalidad de comercializarse.

#### **4.2.3 Efectos externos**

- ✓ Se contará con una herramienta que posibilitará a los usuarios tener acceso a la información importante de una manera más segura y rápida.

#### **4.2.4 Intangibles**

En la evaluación económica siempre hay componentes sensibles por una comunidad que provoca desventaja o beneficio, pero en el instante de convertirlas en unidades monetarias resulta poco práctico y a la vez imposible. Estos efectos intangibles tienen la manera de comprobar el rigor de los efectos a través de los beneficios y costos del proyecto. (21)

**Los costos** nos brindan firmeza al cambio.

**Los Beneficios** por otro lado facilitan:

- Mayor comodidad además de una organización excelente de la información para los usuarios.
- Mejora en la calidad y visibilidad de la información.
- Facilidad a la hora de buscar.
- Menor tiempo empleado en el proceso de consulta de los datos.

### **4.3 Ficha de Costo**

Para calcular el costo económico del proyecto se empleará el procedimiento para confeccionar una Ficha de Costo de un Producto Informático. La ficha contiene los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional.

#### **Costos en Moneda Librementemente Convertible**

##### **Costos Directos.**

1. Compra de equipos de cómputo: No procede.
2. Alquiler de equipos de cómputo: No procede.
3. Compra de licencia de Software: No procede.
4. Depreciación de equipos: 64,34.
5. Materiales directos: No procede.

Subtotal: \$64,34.

##### **Costos Indirectos.**

1. Formación del personal que elabora el proyecto: No procede.
2. Gastos en llamadas telefónicas: No procede.
3. Gastos para el mantenimiento del centro: No procede.
4. Know How: No procede.
5. Gastos en representación: No procede.

Subtotal: \$0.00.

##### **Gastos de distribución y venta.**

1. Participación en ferias o exposiciones: No procede.

2. Gastos en transportación: No procede.

3. Compra de materiales de propagandas: No procede.

Subtotal: \$0.00.

Total: \$64,34 CUC

### **Costos en Moneda Nacional**

✓ Costos Directos.

1. Salario del personal que laborará en el proyecto: \$100.00.

2. El 5% del total de gastos por salarios se dedica a la seguridad social: No procede.

3. El 0.09% del salario total, por concepto de vacaciones a acumular: No procede.

4. Gasto por consumo de energía eléctrica: \$ 35.20.

5. Gastos en llamadas telefónicas: No procede.

6. Gastos administrativos: No procede.

Total: \$135.20 CUP

De acuerdo a lo antes expuesto, la técnica que se seleccionó para la evaluación de factibilidad del proyecto es la metodología Costo-Efectividad. Esta Metodología está conformada por la técnica de punto de equilibrio ajustada a proyectos donde los beneficios tangibles no son tan indiscutibles, el análisis se fundamenta solamente en los costos. Para esta técnica es necesario definir una variable discreta que varíe los costos. Teniendo en cuenta que el costo para este proyecto es despreciable tomaremos como costo, el tiempo en minutos empleado para realizar las actividades en la VRIP, la variable sería la complejidad de las pruebas que se desarrollan durante este proceso.

### **Valores de la variable (Solución manual):**

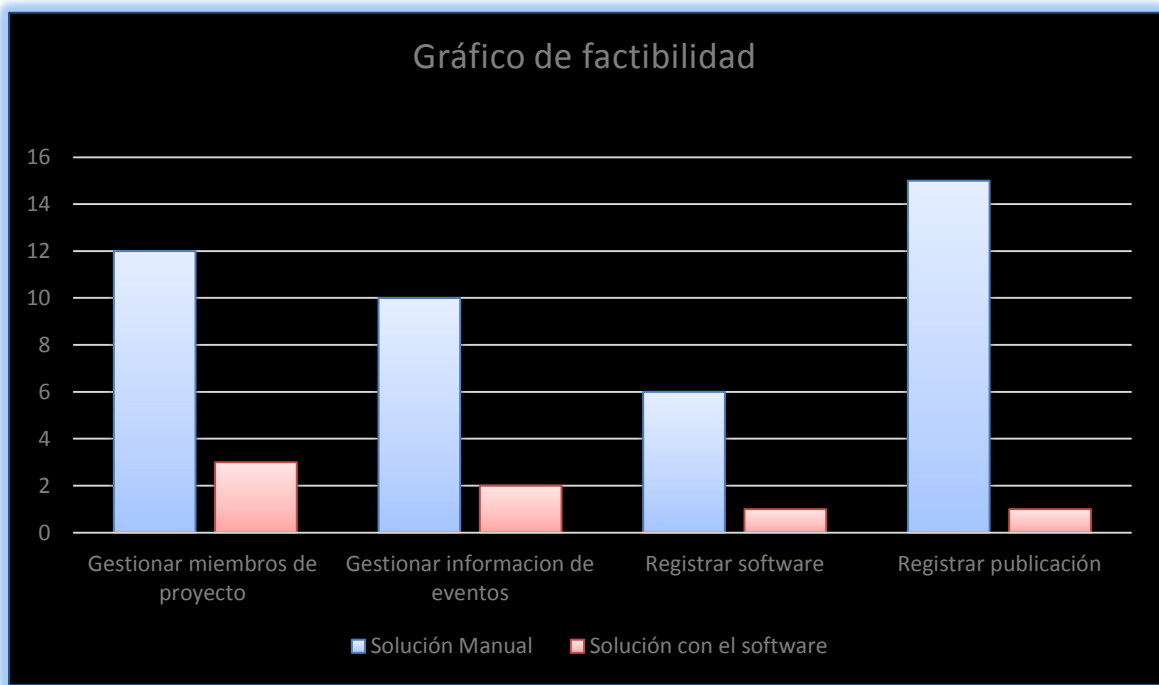
1. Gestionar miembros de proyecto (12 min).



2. Gestionar información de eventos. (10 min).
3. Registrar software. (6 min).
4. Registrar publicación (15 min).

**Valores de la variable (Solución con el software):**

1. Gestionar miembros de proyecto (3 min).
2. Gestionar información de eventos. (2 min).
3. Registrar software. (1 min).
4. Registrar publicación (1 min).



**Figura 5 Gráfico relación solución manual y solución con el software.**

Teniendo en cuenta los resultados reflejados en la gráfica en cuanto al Punto de Equilibrio queda demostrada la factibilidad del sistema evidenciado por la relación entre la complejidad del problema (cantidad de variables) y el tiempo que demora la solución del mismo de forma manual y automatizada.

### ***Conclusiones del Capítulo***

En este capítulo se realizó el estudio de factibilidad del proyecto mediante la Metodología Costo-Efectividad (Beneficio). En este se analizaron los efectos económicos, los beneficios y costos intangibles, y además se calculó el costo de ejecución del proyecto mediante la ficha de costo, la cual arrojó como resultado \$ 64,34 CUC y \$ 100,00 CUP por mes, teniendo en cuenta que el software se desarrolló en un período de 3 meses entonces el costo total en CUP es de \$ 300,00, y \$193,02 en CUC, demostrándose así la factibilidad del proyecto.

## **CONCLUSIONES GENERALES**

Con el desarrollo de la aplicación web para la gestión de la información de la Vicerrectoría de Investigación y Postgrado del ISMMM se dio cumplimiento a los objetivos trazados en esta investigación, pues se obtuvo como resultado una herramienta informática en la que se aplican los resultados de la investigación realizada arribándose a las siguientes conclusiones:

- ✓ Se elaboró el marco teórico metodológico de la investigación, permitiendo esto que quedara identificada la situación problemática existente, además de los fundamentos para comenzar con el diseño e implementación de la aplicación.
- ✓ Se realizó un estudio de los procesos que constituyen la actividad científica y la innovación tecnológica en la Vicerrectoría de Investigación y Postgrado del ISMMM.
- ✓ Durante la implementación del sistema, siguiendo la metodología XP se generaron 49 requisitos funcionales, 16 historias de usuario, 14 tarjetas CRC, 16 pruebas de aceptación y 49 tareas de ingeniería.
- ✓ Se realizó el estudio de factibilidad económica, y operativa del software; arrojando resultados satisfactorios para el proyecto.

Por lo antes expuesto se concluye que los objetivos propuestos fueron cumplidos satisfactoriamente.

## **RECOMENDACIONES DEL AUTOR**

El autor de la presente investigación ha determinado ofrecer las siguientes recomendaciones:

- Poner en funcionamiento el sistema.
- La continuidad y profundización de la investigación sobre el proceso de la actividad científica e innovación tecnológica en el Instituto Superior Minero Metalúrgico de Moa (ISMMM) a través de la herramienta informática que permite obtener la información deseada con mayor rapidez.
- La introducción y generalización en la práctica de la herramienta informática para gestionar la información de los procesos que constituyen la actividad científica y la innovación tecnológica en el Instituto Superior Minero Metalúrgico de Moa (ISMMM).
- Que se implementen más funcionalidades al sistema.

## REFERENCIAS BIBLIOGRÁFICAS

1. Comunidad dotProject. Software de Administración de proyectos dotproject. [En línea] 2014. <http://www.dotproject.net>.
2. RedMine.org. [En línea] 2014. [www.redmine.org](http://www.redmine.org).
3. **Molina, Ernesto Yariel Gámez.** *Sistema Informático para el Control de Inventario de Equipos de Cómputo en la Empresa Agropecuaria Sagua de Tánamo.* 2012.
4. **Pupo, Ing. Ricardo Salazar.** *Sistema de Gestión de Capacitación ECRIN.* Moa : ISMMM, 2013.
5. **Silva, Ing. Exneider Proenza.** *Sistema de Costos de Calidad para la Empresa Empleadora del níquel.* Moa : ISMMM, 2010.
6. KI Networks: Achour, Adnane. [En línea] <http://researchnetworks.ki.se/converis/person/262>.
7. **Santiago.** *Programación web II.* s.l. : I.S.C.B.M, 2009.
8. MySQL. [En línea] 15 de 04 de 2014. <http://Fsinemed.com%2Frecursos%2Fdocs%2FMySQL.pdf>.
9. **Pressman, Roger.** *Ingeniería del Software, un enfoque práctico.* s.l. : McGraw - Hill, 2011.
10. **Jeffries, R.** What is Extreme programming. [En línea] [Citado el: 12 de 04 de 2014.] [www.xpprogramming.com/xpmag/whatisxp.htm](http://www.xpprogramming.com/xpmag/whatisxp.htm).
11. **Escribano, Gerardo Fernandez.** *Presentación XP.pdf.* 2002.
12. **Gutiérrez, Javier J.** *¿Qué es un framework web?* [pdf]
13. —. *¿Qué es un framework Web?* [En línea] [Citado el: 10 de 06 de 2014.] [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf).
14. **Gómez, Edgar J.** *¿Qué es un framework? Servicios Informáticos.* [En línea] [Citado el: 11 de 06 de 2014.] <http://edgargomez.es/que-es-un-framework/>.
15. **Pablo Martínez, Pablo Ruiz Díaz, Sebastián Waisbrot.** *Manual de CodeIgniter en español.* [pdf]
16. Programación en castellano. [En línea] [Citado el: 10 de 06 de 2014.] [http://www.programacion.com/articulo/distintos\\_framework\\_para\\_php\\_379](http://www.programacion.com/articulo/distintos_framework_para_php_379).

17. **GNUChile, Fundación.** *Taller Introducción a Yii Framework.* [pdf] Chile : Fundación de Desarrollo Tecnológico Libre.
18. **LLC, Yii Software.** Yii Framework-Tutorials. [En línea] [Citado el: 09 de 06 de 2014.] <http://www.yiiframework.com/doc/guide/1.1/es/quickstart.what-is-yii>.
19. ER-Studio. [En línea] [Citado el: 20 de 04 de 14.] <http://www.monografias.com/trabajos14/modelodebase/modelodebase.shtml..>
20. EmbarcaderoER/Studio. [En línea] [Citado el: 21 de 05 de 14.] <http://bureaudeprensa.com/es/..>
21. **Falcón, Ing. Yuniel Suárez.** *SGI para la estimación y mitigación de riesgos de desastres para la provincia Holguín.* Moa : ISMMM, 2012.
22. **Bañeres, Juan Palacio.** Gestión ágil de proyectos: Scrum. *Navegapolis.net.* [En línea] <http://www.navegapolis.net>.
23. Wikipedia. [En línea] [www.wikipedia.org](http://www.wikipedia.org).
24. Curso en castellano de arquitecturas de software. [En línea] 2012. <http://www.sceu.frba.utn.edu.ar/e-learning/cursos-a-distancia/Inform%C3%A1tica/Arquitecturas-de-Software/temario>.
25. **Chile, Fundación GNU.** Taller introducción Framework Yii. [En línea] <http://www.fundaciongnuchile.cl>.
26. **Llorente, Yisel Ochoa.** *Sistema Informatizado Para El Control De Medios Informáticos Y De Comunicaciones Para La UEBDC "Ruben D. Suarez Abella".* 2012.
27. **Duarte, Gabriel.** Guía básica de yii framework. [En línea] 10 de 2011. [Citado el: 20 de 02 de 2014.] <http://leninmhs.files.wordpress.com/2011/10/guia-basica-de-yii-framework.pdf>.

## BIBLIOGRAFÍAS

1. Comunidad dotProject. Software de Administración de proyectos dotproject. [En línea] 2014. <http://www.dotproject.net>.
2. RedMine.org. [En línea] 2014. [www.redmine.org](http://www.redmine.org).
3. **Molina, Ernesto Yariel Gámez.** *Sistema Informático para el Control de Inventario de Equipos de Cómputo en la Empresa Agropecuaria Sagua de Tánamo.* 2012.
4. **Pupo, Ing. Ricardo Salazar.** *Sistema de Gestión de Capacitación ECRIN.* Moa : ISMMM, 2013.
5. **Silva, Ing. Exneider Proenza.** *Sistema de Costos de Calidad para la Empresa Empleadora del níquel.* Moa : ISMMM, 2010.
6. KI Networks: Achour, Adnane. [En línea] <http://researchnetworks.ki.se/converis/person/262>.
7. **Santiago.** *Programación web II.* s.l. : I.S.C.B.M, 2009.
8. MySQL. [En línea] 15 de 04 de 2014. <http://Fsinemed.com%2Frecursos%2Fdocs%2FMySQL.pdf>.
9. **Pressman, Roger.** *Ingeniería del Software, un enfoque práctico.* s.l. : McGraw - Hill, 2011.
10. **Jeffries, R.** What is Extreme programming. [En línea] [Citado el: 12 de 04 de 2014.] [www.xpprogramming.com/xpmag/whatisxp.htm](http://www.xpprogramming.com/xpmag/whatisxp.htm).
11. **Escribano, Gerardo Fernandez.** *Presentación XP.pdf.* 2002.
12. **Gutiérrez, Javier J.** *¿Qué es un framework web?* [pdf]
13. —. *¿Qué es un framework Web?* [En línea] [Citado el: 10 de 06 de 2014.] [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf).
14. **Gómez, Edgar J.** *¿Qué es un framework? Servicios Informáticos.* [En línea] [Citado el: 11 de 06 de 2014.] <http://edgargomez.es/que-es-un-framework/>.
15. **Pablo Martínez, Pablo Ruiz Díaz, Sebastián Waisbrot.** *Manual de CodeIgniter en español.* [pdf]
16. Programación en castellano. [En línea] [Citado el: 10 de 06 de 2014.] [http://www.programacion.com/articulo/distintos\\_framework\\_para\\_php\\_379](http://www.programacion.com/articulo/distintos_framework_para_php_379).

17. **GNUChile, Fundación.** *Taller Introducción a Yii Framework.* [pdf] Chile : Fundación de Desarrollo Tecnológico Libre.
18. **LLC, Yii Software.** Yii Framework-Tutorials. [En línea] [Citado el: 09 de 06 de 2014.] <http://www.yiiframework.com/doc/guide/1.1/es/quickstart.what-is-yii>.
19. ER-Studio. [En línea] [Citado el: 20 de 04 de 14.] <http://www.monografias.com/trabajos14/modelodebase/modelodebase.shtml..>
20. EmbarcaderoER/Studio. [En línea] [Citado el: 21 de 05 de 14.] <http://bureaudeprensa.com/es/..>
21. **Falcón, Ing. Yuniel Suárez.** *SGI para la estimación y mitigación de riesgos de desastres para la provincia Holguín.* Moa : ISMMM, 2012.
22. **Bañeres, Juan Palacio.** Gestión ágil de proyectos: Scrum. *Navegapolis.net.* [En línea] <http://www.navegapolis.net>.
23. Wikipedia. [En línea] [www.wikipedia.org](http://www.wikipedia.org).
24. Curso en castellano de arquitecturas de software. [En línea] 2012. <http://www.sceu.frba.utn.edu.ar/e-learning/cursos-a-distancia/Inform%C3%A1tica/Arquitecturas-de-Software/temario>.
25. **Chile, Fundación GNU.** Taller introducción Framework Yii. [En línea] <http://www.fundaciongnuchile.cl>.
26. **Duarte, Gabriel.** Guía básica de yiiframework. [En línea] 10 de 2011. [Citado el: 20 de 02 de 2014.] <http://leninmhs.files.wordpress.com/2011/10/guia-basica-de-yii-framework.pdf>.
27. **Llorente, Yisel Ochoa.** *SISTEMA INFORMATIZADO PARA EL CONTROL DE MEDIOS INFORMÁTICOS Y DE COMUNICACIONES PARA LA UEBDC "RUBEN D. SUAREZ ABELLA".* 2012.



## **GLOSARIO DE TÉRMINOS**

**Actividad Científica:** Desarrollo del pensamiento estratégico para la creación de objetivos y la ejecución de las acciones para lograrlos. Obtención del conocimiento mediante la observación de patrones regulares, de razonamientos y de experimentación en ámbitos específicos, a partir de los cuales se generan preguntas, se construyen hipótesis, se deducen principios y se elaboran leyes generales y esquemas metódicamente organizados.

**Actualización:** Cambios que se aplican a un programa, para corregir errores, agregarle funcionalidades, etc.

**Análisis costo beneficio:** El coste-beneficio es una lógica o razonamiento basado en el principio de obtener los mayores y mejores resultados al menor esfuerzo invertido, tanto por eficiencia técnica como por motivación humana. Se supone que todos los hechos y actos pueden evaluarse bajo esta lógica, aquellos dónde los beneficios superan el coste son exitosos, caso contrario fracasan.

**Aplicación Web:** Una aplicación web es un sistema informático que los usuarios usan accediendo a un servidor web a través de los protocolos de Internet. Las aplicaciones web son populares por su practicidad del navegador web como cliente ligero.

**API:** Una API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

**ASP:** Active Server Pages, también conocido como ASP clásico, es una tecnología de Microsoft para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Services (IIS).

**Ajax:** Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas

sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

**Beneficio:** Ganancia económica que se obtiene de un negocio, inversión u otra actividad mercantil.

**Bugs:** Errores que se presentan en la aplicación que atentan contra su correcto funcionamiento.

**Cliente:** Persona, organización o grupo de personas que solicita la construcción de un sistema, ya sea empezando desde cero, o mediante el refinamiento de versiones sucesivas.

**Costo:** En economía el coste o costo es el valor monetario de los consumos de factores que supone el ejercicio de una actividad económica destinada a la producción de un bien o servicio.

**Compatibilidad:** Condición que hace que un programa y un sistema, arquitectura o aplicación logren comprenderse correctamente tanto directamente o indirectamente mediante un algoritmo.

**Concurrencia:** Coincidencia, concurso simultáneo de varias circunstancias. Es la simultaneidad en la ejecución de múltiples tareas interactivas. Estas tareas pueden ser un conjunto de procesos o hilos de ejecución creados por un único programa.

**Gestión de información:** Es un conjunto de procesos por los cuales se controla el ciclo de vida de la información, desde su obtención - por creación o captura, hasta su disposición final - archivada o eliminada. Los procesos también comprenden la extracción, combinación, depuración y distribución de la información a los interesados.

**Herramientas:** Son los ambientes de apoyo necesario para automatizar las prácticas de Ingeniería de Software.

**IDE:** Integrated Development Environment / Entorno de Desarrollo Integrado. Entorno de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

**Innovación tecnológica:** Actividad cuyo resultado sea un avance tecnológico en la obtención de nuevos productos o procesos de producción o mejoras sustanciales de los ya existentes.

**Interfaz:** Conjunto de representaciones de operaciones públicas.

**Iteraciones:** En el contexto de un proyecto se refieren a la técnica de desarrollar y entregar componentes incrementales de funcionalidades de un negocio. Una iteración resulta en uno o más paquetes atómicos y completos del trabajo del proyecto que pueda realizar alguna función tangible del negocio.

Múltiples iteraciones contribuyen a crear un producto completamente integrado.

**JavaScript:** Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

**Licencia BSD:** Es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

**Lógica del negocio:** Parte de un sistema que se encarga de las tareas relacionadas con los procesos de un negocio, tales como ventas, control de inventario, contabilidad, etc.

**Multiplataforma:** Es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas o sistemas operativos.

**MES:** Ministerio de la Educación Superior.

**Metodología Ágil:** Constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de proyectos que las metodologías convencionales debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

**Metodología de desarrollo:** Se define como un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información.

**Metodologías tradicionales:** Metodologías basadas en procesos.

**Navegador web:** Un navegador o navegador web (del inglés, web browser) es una aplicación que opera a través de Internet, interpretando la información de archivos y sitios web para que podamos ser capaces de leerla, (ya se encuentre ésta alojada en un servidor dentro de la World Wide Web o en un servidor local).

**POO:** Programación Orientada a Objetos.

**Proceso:** Conjunto de las fases sucesivas de un fenómeno natural o de una operación artificial. *INF.* Programa en ejecución.

**Proyecto:** Esfuerzo de desarrollo para llevar un sistema a lo largo de un ciclo de vida.

**Programación extrema:** Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

**Requisitos:** Son las funciones, servicios y restricciones operativas del sistema.

**Script:** En informática un guión, archivo de órdenes o archivo de procesamiento por lotes, es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. Los guiones son casi siempre interpretados, pero no todo programa interpretado es considerado un guión.

**Sistema:** Delimita el mundo sobre el cual se está construyendo el modelo

**SQLite:** es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (~275 kB) biblioteca en C. SQLite es un proyecto de dominio público

**Servidor Web:** Un programa que corre sobre el servidor que escucha las peticiones HTTP (Hyper Text Markup Language) que le llegan y las satisface. Dependiendo del tipo de petición, el servidor web buscará una página web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición

**Testing:** Proceso de pruebas usado para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador o videojuego. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

**VRIP:** Vicerrectoría de Investigación y Postgrado.

## **Anexo 1 Historias de Usuario**

**Tabla 2.6 HU No. 1 Autenticar Usuario**

| Historia de usuario   |                             |
|---|-----------------------------|
| Número: 1   | Usuario: Cliente            |
| Nombre: Autenticar usuarios.  |                             |
| Prioridad en el negocio: Alta   | Riesgo en desarrollo: media |
| Puntos estimados: 1   | Iteración asignada: Primera |
| Programador responsable: Deysi Columbié Hernández   |                             |
| Descripción: Los usuarios ingresan sus datos para entrar al sistema (nombre de usuario y contraseña). El sistema verifica que los datos estén correctos, de no ser correctos el usuario no podrá entrar al sistema. |                             |
| Observaciones: Confirmado por el cliente.   |                             |

**Tabla 2.7 HU No. 2 Gestionar Usuario**

| Historia de usuario   |                             |
|---|-----------------------------|
| Número: 2   | Usuario: Administrador      |
| Nombre: Gestionar usuarios.   |                             |
| Prioridad en el negocio: Alta   | Riesgo en desarrollo: Alta  |
| Puntos estimados: 1   | Iteración asignada: Primera |
| Programador responsable: Deysi Columbié Hernández   |                             |
| Descripción: el usuario debe entrar los datos requeridos después podrá realizar las operaciones correspondientes. |                             |

Observaciones: Confirmado por el cliente.

**Tabla 2.8 HU No. 3 Gestionar Línea de Investigación**

| Historia de usuario   |                             |
|---|-----------------------------|
| Número: 3   | Usuario: Administrador      |
| Nombre: Gestionar línea de investigación.   |                             |
| Prioridad en el negocio: Alta   | Riesgo en desarrollo: Alta  |
| Puntos estimados: 1   | Iteración asignada: Primera |
| Programador responsable: Deysi Columbié Hernández   |                             |
| Descripción: el usuario debe entrar los datos requeridos después podrá realizar las operaciones correspondientes. |                             |
| Observaciones: Confirmado por el cliente.   |                             |

**Tabla 2.9 HU No. 4 Gestionar indicadores dinámicos de la evaluación**

| Historia de usuario   |                             |
|---|-----------------------------|
| Número: 4   | Usuario: Administrador      |
| Nombre: Gestionar indicadores dinámicos de la evaluación.   |                             |
| Prioridad en el negocio: Alta   | Riesgo en desarrollo: Alta  |
| Puntos estimados: 1   | Iteración asignada: Primera |
| Programador responsable: Deysi Columbié Hernández   |                             |
| Descripción: el usuario debe entrar los datos requeridos después podrá realizar las operaciones correspondientes. |                             |

Observaciones: Confirmado por el cliente.

**Tabla 2.10 HU No. 5 Gestionar proyectos**

| Historia de usuario   |                             |
|---|-----------------------------|
| Número: 5   | Usuario: Administrador      |
| Nombre: Gestionar Proyectos.  |                             |
| Prioridad en el negocio: Alta   | Riesgo en desarrollo: Alta  |
| Puntos estimados: 1   | Iteración asignada: Primera |
| Programador responsable: Deysi Columbié Hernández   |                             |
| Descripción: el usuario debe entrar los datos requeridos después podrá realizar las operaciones correspondientes. |                             |
| Observaciones: Confirmado por el cliente.   |                             |

**Tabla 2.11 HU No. 6 Gestionar secuencia de actividades.**

| Historia de usuario   |                                    |
|---|------------------------------------|
| Número: 6   | Usuario: administrador, secretaria |
| Nombre: Gestionar secuencia de actividades.   |                                    |
| Prioridad en el negocio: Alta   | Riesgo en desarrollo: Alta         |
| Puntos estimados: 1   | Iteración asignada: Primera        |
| Programador responsable: Deysi Columbié Hernández                                   |                                    |
| Descripción: el usuario debe entrar los datos requeridos después podrá realizar las |                                    |



|   |
|---|
| operaciones correspondientes.             |
| Observaciones: Confirmado por el cliente. |

**Tabla 2.12 HU No. 7 Gestionar miembro del proyecto**

| Historia de usuario   |                             |
|---|-----------------------------|
| Número: 7   | Usuario: Administrador      |
| Nombre: Gestionar miembro del proyecto.   |                             |
| Prioridad en el negocio: Alta   | Riesgo en desarrollo: Alta  |
| Puntos estimados: 1   | Iteración asignada: Segunda |
| Programador responsable: Deysi Columbié Hernández   |                             |
| Descripción: el usuario debe entrar los datos requeridos después podrá realizar las operaciones correspondientes. |                             |
| Observaciones: Confirmado por el cliente.   |                             |

**Tabla 2.13 HU No. 8 Gestionar documentos**

| Historia de usuario           |                             |
|-------------------------------|-----------------------------|
| Número: 8                     | Usuario: Administrador      |
| Nombre: Gestionar documentos. |                             |
| Prioridad en el negocio: Alta | Riesgo en desarrollo: Alta  |
| Puntos estimados: 1           | Iteración asignada: Segunda |

|   |
|---|
| Programador responsable: Deysi Columbié Hernández   |
| Descripción: el usuario debe entrar los datos requeridos después podrá realizar las operaciones correspondientes. |
| Observaciones: Confirmado por el cliente.   |

**Tabla 2.14 HU No. 9 Gestionar plan de resultados**

|   |                                    |
|---|------------------------------------|
| Historia de usuario   |                                    |
| Número: 9   | Usuario: administrador, secretaria |
| Nombre: Mostrar alerta de vencimiento del pasaporte.  |                                    |
| Prioridad en el negocio: Alta   | Riesgo en desarrollo: Alta         |
| Puntos estimados: 1   | Iteración asignada: Segunda        |
| Programador responsable: Deysi Columbié Hernández   |                                    |
| Descripción: el usuario debe entrar los datos requeridos después podrá realizar las operaciones correspondientes. |                                    |
| Observaciones: Confirmado por el cliente.   |                                    |

**Tabla 2.15 HU No. 10 Registrar software.**

|                               |                                    |
|-------------------------------|------------------------------------|
| Historia de usuario           |                                    |
| Número: 10                    | Usuario: administrador, secretaria |
| Nombre: Registrar software.   |                                    |
| Prioridad en el negocio: Alta | Riesgo en desarrollo: Alta         |
| Puntos estimados: 1           | Iteración asignada: Segunda        |

|  |
|--|
| Programador responsable: Deysi Columbié Hernández  |
| Descripción: el usuario introduce los datos requeridos, luego podrá realizar las operaciones correspondientes. |
| Observaciones: Confirmado por el cliente.  |

**Tabla 2.16 HU No. 11 Listar registros de software**

|   |  |  |  |
|---|--|--|--|
| <b>Número: 11</b>   |  | <b>Usuario: Listar registros de software</b> |  |
| Nombre: Listar registro de software.                                    |  |  |  |
| Prioridad en el negocio: Media  |  | Riesgo en desarrollo: Media                  |  |
| Puntos estimados: 1   |  | Iteración asignada: Segunda                  |  |
| Programador responsable: Deysi Columbié Hernández                       |  |  |  |
| Descripción: el sistema muestra los datos de los registros de software. |  |  |  |
| Observaciones: Confirmado por el cliente.                               |  |  |  |

**Tabla 2.17 HU No. 12 Gestionar artículo.**

|  |                                    |
|--|------------------------------------|
| <b>Historia de usuario</b>   |                                    |
| Número: 12   | Usuario: administrador, secretaria |
| Nombre: Gestionar artículo.  |                                    |
| Prioridad en el negocio: Media   | Riesgo en desarrollo: Media        |
| Puntos estimados: 1  | Iteración asignada: Tercera        |
| Programador responsable: Deysi Columbié Hernández  |                                    |
| Descripción: el usuario introduce los datos requeridos, luego podrá realizar las operaciones |                                    |

|   |
|---|
| correspondientes.                         |
| Observaciones: Confirmado por el cliente. |

**Tabla 2.18 HU No. 13 Gestionar evento**

| Historia de usuario   |                             |
|---|-----------------------------|
| Número: 13  | Usuario: Administrador      |
| Nombre: Gestionar evento.   |                             |
| Prioridad en el negocio: Media  | Riesgo en desarrollo: Media |
| Puntos estimados: 1   | Iteración asignada: Tercera |
| Programador responsable: Deysi Columbié Hernández   |                             |
| Descripción: el usuario introduce los datos requeridos, luego podrá realizar las operaciones correspondientes |                             |
| Observaciones: Confirmado por el cliente.   |                             |

**Tabla 2.19 HU No. 14 Gestionar premios**

| Historia de usuario  |                                    |
|--|------------------------------------|
| Número: 14   | Usuario: administrador, secretaria |
| Nombre: Gestionar premios.   |                                    |
| Prioridad en el negocio: Media   | Riesgo en desarrollo: Media        |
| Puntos estimados: 1  | Iteración asignada: Tercera        |
| Programador responsable: Deysi Columbié Hernández  |                                    |
| Descripción: el usuario introduce los datos requeridos, luego podrá realizar las operaciones |                                    |

|   |
|---|
| correspondientes                          |
| Observaciones: Confirmado por el cliente. |

**Tabla 2.19 HU No. 15 Registrar publicación.**

| Historia de usuario   |                                    |
|---|------------------------------------|
| Número: 15  | Usuario: administrador, secretaria |
| Nombre: Registrar publicación.  |                                    |
| Prioridad en el negocio: Media  | Riesgo en desarrollo: Media        |
| Puntos estimados: 1   | Iteración asignada: Tercera        |
| Programador responsable: Deysi Columbié Hernández   |                                    |
| Descripción: el usuario introduce los datos requeridos, luego podrá realizar las operaciones correspondientes |                                    |
| Observaciones: Confirmado por el cliente.   |                                    |

**Tabla 2.20 HU No. 16 Listar registro de publicación.**

| Historia de usuario                               |                             |
|---|-----------------------------|
| Número: 16  | Usuario: Administrador      |
| Nombre: Listar registro de publicación.           |                             |
| Prioridad en el negocio: Alta                     | Riesgo en desarrollo: Alta  |
| Puntos estimados: 1                               | Iteración asignada: Tercera |
| Programador responsable: Deysi Columbié Hernández |                             |

Descripción: el sistema muestra los datos de los registros de publicación.

Observaciones: Confirmado por el cliente.

## Anexo 2 Tarjetas CRC

Tabla 2.21 Tarjeta CRC No. 1

| Nombre de la Clase: Autenticar Usuario |               |
|--|---------------|
| Tipo de clase:                         |               |
| Lógica del negocio                     |               |
| Responsabilidades                      | Colaboradores |
| Autentica al usuario en el sistema.    | login         |

Tabla 2.22 Tarjeta CRC No. 2

| Nombre de la Clase: Gestionar Usuario   |                   |
|---|-------------------|
| Tipo de clase:                          |                   |
| Lógica del negocio                      |                   |
| Responsabilidades                       | Colaboradores     |
| Inserta datos del usuario               | Insertar_usuario  |
| Muestra los datos del usuario           | Mostrar_usuario   |
| Modifica los datos del usuario          | Modificar_usuario |
| Eliminar el usuario de la base de datos | Eliminar_usuario  |

Tabla 2.23 Tarjeta CRC No. 4

| Nombre de la Clase: Gestionar Indicadores dinámicos de la evaluación |
|--|
| Tipo de clase:   |
| Lógica del negocio   |

|   |               |
|---|---------------|
| Responsabilidades                                   | Colaboradores |
| Inserta los indicadores dinámicos de la evaluación  | Insertar_ide  |
| Muestra indicadores dinámicos de la evaluación      | Mostrar_ide   |
| Modifica los indicadores dinámicos de la evaluación | Modificar_ide |
| Eliminar indicadores dinámicos de la evaluación     | Eliminar_ide  |

**Tabla 2.24 Tarjeta CRC No. 5**

|  |                     |
|--|---------------------|
| Nombre de la Clase: Gestionar Proyectos  |                     |
| Tipo de clase:                           |                     |
| Lógica del negocio                       |                     |
| Responsabilidades                        | Colaboradores       |
| Inserta datos del proyecto               | Insertar_ proyecto  |
| Muestra los datos del proyecto           | Mostrar_ proyecto   |
| Modifica los datos del proyecto          | Modificar_ proyecto |
| Eliminar el proyecto de la base de datos | Eliminar_ proyecto  |

**Tabla 2.25 Tarjeta CRC No. 6**

|  |                          |
|--|--------------------------|
| Nombre de la Clase: Gestionar secuencia de actividades |                          |
| Tipo de clase:   |                          |
| Lógica del negocio                                     |                          |
| Responsabilidades                                      | Colaboradores            |
| Inserta datos de la secuencia de actividades           | Insertar_ secuencia_act  |
| Muestra los datos de la secuencia de actividades       | Mostrar_ secuencia_act   |
| Modifica los datos de la secuencia de actividades      | Modificar_ secuencia_act |



|   |                         |
|---|-------------------------|
| Eliminar la secuencia de la base de datos | Eliminar_ secuencia_act |
|---|-------------------------|

**Tabla 2.26 Tarjeta CRC No. 7**

|   |                    |
|---|--------------------|
| Nombre de la Clase: Gestionar Miembro de Proyecto   |                    |
| Tipo de clase:                                      |                    |
| Lógica del negocio                                  |                    |
| Responsabilidades                                   | Colaboradores      |
| Inserta datos del Miembro de proyecto               | Insertar_ miembro  |
| Muestra los datos del Miembro de proyecto           | Mostrar_ miembro   |
| Modifica los datos del Miembro de proyecto          | Modificar_ miembro |
| Eliminar el Miembro de proyecto de la base de datos | Eliminar_ miembro  |

**Tabla 2.27 Tarjeta CRC No. 8**

|   |                      |
|---|----------------------|
| Nombre de la Clase: Gestionar Documentos  |                      |
| Tipo de clase:                            |                      |
| Lógica del negocio                        |                      |
| Responsabilidades                         | Colaboradores        |
| Inserta datos del documento               | Insertar_ documento  |
| Muestra los datos del documento           | Mostrar_ documento   |
| Modifica los datos del documento          | Modificar_ documento |
| Eliminar el documento de la base de datos | Eliminar_ documento  |

**Tabla 2.28 Tarjeta CRC No. 9**

|   |
|---|
| Nombre de la Clase: Gestionar plan de resultado |
|---|

|   |                 |
|---|-----------------|
| Tipo de clase:                                    |                 |
| Lógica del negocio                                |                 |
| Responsabilidades                                 | Colaboradores   |
| Inserta datos del plan de resultado               | Insertar_ plan  |
| Muestra los datos del plan de resultado           | Mostrar_ plan   |
| Modifica los datos del plan de resultado          | Modificar_ plan |
| Eliminar el plan de resultado de la base de datos | Eliminar_ plan  |

**Tabla 2.29 Tarjeta CRC No. 10**

|   |                    |
|---|--------------------|
| <b>Nombre de la Clase: Registrar software</b>   |                    |
| Tipo de clase:                                  |                    |
| Lógica del negocio                              |                    |
| Responsabilidades                               | Colaboradores      |
| Inserta datos del software                      | Insertar_ registro |
| Muestra un listado de los registros de software | Mostrar_ registro  |

**Tabla 2.30 Tarjeta CRC No. 11**

|   |                     |
|---|---------------------|
| <b>Nombre de la Clase: Gestionar Artículo</b> |                     |
| Tipo de clase:                                |                     |
| Lógica del negocio                            |                     |
| Responsabilidades                             | Colaboradores       |
| Inserta datos del artículo                    | Insertar_ articulo  |
| Muestra los datos del artículo                | Mostrar_ articulo   |
| Modifica los datos del artículo               | Modificar_ articulo |

|  |                    |
|--|--------------------|
| Eliminar el artículo de la base de datos | Eliminar_ articulo |
|--|--------------------|

**Tabla 2.31 Tarjeta CRC No. 12**

| Nombre de la Clase: Gestionar Evento   |                   |
|--|-------------------|
| Tipo de clase:                         |                   |
| Lógica del negocio                     |                   |
| Responsabilidades                      | Colaboradores     |
| Inserta datos del evento               | Insertar_ evento  |
| Muestra los datos del evento           | Mostrar_ evento   |
| Modifica los datos del evento          | Modificar_ evento |
| Eliminar el evento de la base de datos | Eliminar_ evento  |

**Tabla 2.32 Tarjeta CRC No. 13**

| Nombre de la Clase: Gestionar Premios |                    |
|---------------------------------------|--------------------|
| Tipo de clase:                        |                    |
| Lógica del negocio                    |                    |
| Responsabilidades                     | Colaboradores      |
| Inserta datos del premio              | Insertar_ premios  |
| Muestra los datos del premio          | Mostrar_ premios   |
| Modifica los datos del premio         | Modificar_ premios |
| Elimina el premio de la base de datos | Eliminar_ premios  |

**Tabla 2.33 Tarjeta CRC No. 14**

| Nombre de la Clase: Registrar publicación |
|---|
|---|

|  |                               |
|--|-------------------------------|
| Tipo de clase:                                       |                               |
| Lógica del negocio                                   |                               |
| Responsabilidades                                    | Colaboradores                 |
| Inserta datos del registro de la publicación         | Registrar_ publicacion        |
| Muestra un listado de los registros de publicaciones | Mostrar_ listado_ publicacion |

## **Anexo 3 Tareas de Ingeniería**

**Tabla 3.6 Tarea de programación Autenticar usuario.**

| <b>Tarea de Programación</b>                             |                       |
|--|-----------------------|
| No. Tarea: 1   | No. Historia: 1       |
| Nombre de la tarea: Autenticar usuarios.                 |                       |
| Tipo de tarea: Desarrollo                                | Puntos estimados: 1   |
| Fecha inicio: 22/01/2014                                 | Fecha fin: 22/01/2014 |
| Programador responsable: Deysi Columbié Hernández        |                       |
| Descripción: Permite autenticar los usuarios al sistema. |                       |

**Tabla 3.7 Tarea de programación Insertar usuario.**

| <b>Tarea de Programación</b>   |                       |
|--|-----------------------|
| No. Tarea: 2   | No. Historia: 2       |
| Nombre de la tarea: Insertar usuarios.   |                       |
| Tipo de tarea: Desarrollo  | Puntos estimados: 0.1 |
| Fecha inicio: 23/01/2014   | Fecha fin: 23/01/2014 |
| Programador responsable: Deysi Columbié Hernández                                |                       |
| Descripción: Permite insertar los usuarios que van a interactuar con el sistema. |                       |

**Tabla 3.8 Tarea de programación Mostrar usuario.**

| <b>Tarea de Programación</b> |                 |
|------------------------------|-----------------|
| No. Tarea: 3                 | No. Historia: 2 |

|   |                       |
|---|-----------------------|
| Nombre de la tarea: Mostrar usuarios.                               |                       |
| Tipo de tarea: Desarrollo   | Puntos estimados: 0.1 |
| Fecha inicio: 24/01/2014  | Fecha fin: 24/01/2014 |
| Programador responsable: Deysi Columbié Hernández                   |                       |
| Descripción: Permite mostrar los datos de los usuarios del sistema. |                       |

**Tabla 3.9 Tarea de programación Modificar usuario.**

|   |                       |
|---|-----------------------|
| Tarea de Programación   |                       |
| No. Tarea: 4  | No. Historia: 2       |
| Nombre de la tarea: Modificar usuarios.                               |                       |
| Tipo de tarea: Desarrollo   | Puntos estimados: 0.2 |
| Fecha inicio: 25/01/2014  | Fecha fin: 26/01/2014 |
| Programador responsable: Deysi Columbié Hernández                     |                       |
| Descripción: Permite modificar los datos de los usuarios del sistema. |                       |

**Tabla 3.10 Tarea de programación Eliminar usuario.**

|  |                       |
|--|-----------------------|
| Tarea de Programación  |                       |
| No. Tarea: 5   | No. Historia: 2       |
| Nombre de la tarea: Eliminar usuarios.                                     |                       |
| Tipo de tarea: Desarrollo  | Puntos estimados: 0.1 |
| Fecha inicio: 27/01/2014   | Fecha fin: 27/01/2014 |
| Programador responsable: Deysi Columbié Hernández                          |                       |
| Descripción: Permite eliminar el usuario seleccionado de la base de datos. |                       |

**Tabla 3.11 Tarea de programación Insertar línea de investigación.**

| Tarea de Programación  |                       |
|--|-----------------------|
| No. Tarea: 6   | No. Historia: 3       |
| Nombre de la tarea: insertar línea de investigación                          |                       |
| Tipo de tarea: Desarrollo  | Puntos estimados: 0.1 |
| Fecha inicio: 20/05/2014   | Fecha fin: 20/05/2014 |
| Programador responsable: Deysi Columbié Hernández                            |                       |
| Descripción: Permite insertar la línea de investigación en la base de datos. |                       |

**Tabla 3.12 Tarea de programación Mostrar línea de investigación.**

| Tarea de Programación  |                       |
|--|-----------------------|
| No. Tarea: 7   | No. Historia: 3       |
| Nombre de la tarea: mostrar línea de investigación                   |                       |
| Tipo de tarea: Desarrollo  | Puntos estimados: 0.1 |
| Fecha inicio: 21/05/2014   | Fecha fin: 21/05/2014 |
| Programador responsable: Deysi Columbié Hernández                    |                       |
| Descripción: Permite mostrar los datos de la línea de investigación. |                       |

**Tabla 3.13 Tarea de programación Modificar línea de investigación.**

| Tarea de Programación                                |                       |
|--|-----------------------|
| No. Tarea: 8   | No. Historia: 3       |
| Nombre de la tarea: modificar línea de investigación |                       |
| Tipo de tarea: Desarrollo                            | Puntos estimados: 0.2 |
| Fecha inicio: 22/05/2014                             | Fecha fin: 23/05/2014 |
| Programador responsable: Deysi Columbié Hernández    |                       |

Descripción: Permite modificar los datos de la línea de investigación.

**Tabla 3.14 Tarea de programación Eliminar línea de investigación**

| Tarea de Programación                                    |                       |
|--|-----------------------|
| No. Tarea: 9   | No. Historia: 3       |
| Nombre de la tarea: eliminar línea de investigación      |                       |
| Tipo de tarea: Desarrollo                                | Puntos estimados: 0.1 |
| Fecha inicio: 24/05/2014                                 | Fecha fin: 24/05/2014 |
| Programador responsable: Deysi Columbié Hernández        |                       |
| Descripción: Permite eliminar la línea de investigación. |                       |

**Tabla 3.15 Tarea de programación Insertar indicadores dinámicos de la evaluación.**

| Tarea de Programación  |                       |
|--|-----------------------|
| No. Tarea: 10  | No. Historia: 4       |
| Nombre de la tarea: insertar indicadores                                 |                       |
| Tipo de tarea: Desarrollo  | Puntos estimados: 0.1 |
| Fecha inicio: 20/04/2014   | Fecha fin: 20/04/2014 |
| Programador responsable: Deysi Columbié Hernández                        |                       |
| Descripción: Permite insertar el indicador dinámico en la base de datos. |                       |

**Tabla 3.16 Tarea de programación Mostrar indicadores dinámicos de la evaluación.**

| Tarea de Programación                 |                 |
|---------------------------------------|-----------------|
| No. Tarea: 11                         | No. Historia: 4 |
| Nombre de la tarea: mostrar indicador |                 |



|   |                       |
|---|-----------------------|
| Tipo de tarea: Desarrollo   | Puntos estimados: 0.1 |
| Fecha inicio: 21/04/2014  | Fecha fin: 21/04/2014 |
| Programador responsable: Deysi Columbié Hernández                       |                       |
| Descripción: Permite mostrar el indicadores dinámicos de la evaluación. |                       |

**Tabla 3.17 Tarea de programación Modificar indicadores dinámicos de la evaluación.**

|   |                       |
|---|-----------------------|
| <b>Tarea de Programación</b>  |                       |
| No. Tarea: 12   | No. Historia: 4       |
| Nombre de la tarea: modificar indicador                                   |                       |
| Tipo de tarea: Desarrollo   | Puntos estimados: 0.2 |
| Fecha inicio: 22/04/2014  | Fecha fin: 23/04/2014 |
| Programador responsable: Deysi Columbié Hernández                         |                       |
| Descripción: Permite modificar el indicadores dinámicos de la evaluación. |                       |

**Tabla 3.18 Tarea de programación Eliminar indicadores dinámicos de la evaluación.**

|   |                       |
|---|-----------------------|
| <b>Tarea de Programación</b>                      |                       |
| No. Tarea: 13                                     | No. Historia: 4       |
| Nombre de la tarea: eliminar indicador            |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 24/04/2014                          | Fecha fin: 24/04/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite eliminar el indicador.       |                       |

**Tabla 3.19 Tarea de programación Insertar proyectos**

| Tarea de Programación  |                       |
|--|-----------------------|
| No. Tarea: 14  | No. Historia: 5       |
| Nombre de la tarea: insertar proyecto                          |                       |
| Tipo de tarea: Desarrollo                                      | Puntos estimados: 0.1 |
| Fecha inicio: 02/02/2014                                       | Fecha fin: 02/02/2014 |
| Programador responsable: Deysi Columbié Hernández              |                       |
| Descripción: Permite insertar el proyecto en la base de datos. |                       |

**Tabla 3.20 Tarea de programación Mostrar proyectos**

| Tarea de Programación                                |                       |
|--|-----------------------|
| No. Tarea: 15  | No. Historia: 6       |
| Nombre de la tarea: mostrar proyecto                 |                       |
| Tipo de tarea: Desarrollo                            | Puntos estimados: 0.1 |
| Fecha inicio: 03/02/2014                             | Fecha fin: 03/02/2014 |
| Programador responsable: Deysi Columbié Hernández    |                       |
| Descripción: Permite mostrar los datos del proyecto. |                       |

**Tabla 3.21 Tarea de programación Modificar proyectos**

| Tarea de Programación                             |                       |
|---|-----------------------|
| No. Tarea: 16                                     | No. Historia: 7       |
| Nombre de la tarea: modificar proyecto.           |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 04/02/2014                          | Fecha fin: 04/02/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |

Descripción: Permite modificar proyecto.

**Tabla 3.22 Tarea de programación Eliminar proyectos**

| Tarea de Programación                             |                       |
|---|-----------------------|
| No. Tarea: 17                                     | No. Historia: 7       |
| Nombre de la tarea: eliminar proyecto.            |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 05/02/2014                          | Fecha fin: 05/02/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite eliminar proyecto.           |                       |

**Tabla 3.23 Tarea de programación Insertar secuencia de actividades**

| Tarea de Programación                                      |                       |
|--|-----------------------|
| No. Tarea: 18  | No. Historia: 7       |
| Nombre de la tarea: insertar secuencia de actividades.     |                       |
| Tipo de tarea: Desarrollo                                  | Puntos estimados: 0.2 |
| Fecha inicio: 06/02/2014                                   | Fecha fin: 07/02/2014 |
| Programador responsable: Deysi Columbié Hernández          |                       |
| Descripción: Permite insertar la secuencia de actividades. |                       |

**Tabla 3.24 Tarea de programación mostrar secuencia de actividades**

| Tarea de Programación                                |                       |
|--|-----------------------|
| No. Tarea: 19  | No. Historia: 7       |
| Nombre de la tarea: mostrar secuencia de actividades |                       |
| Tipo de tarea: Desarrollo                            | Puntos estimados: 0.1 |

|  |                       |
|--|-----------------------|
| Fecha inicio: 09/02/2014                               | Fecha fin: 09/02/2014 |
| Programador responsable: Deysi Columbié Hernández      |                       |
| Descripción: Permite mostrar secuencia de actividades. |                       |

**Tabla 3.25 Tarea de programación modificar secuencia de actividades**

| Tarea de Programación  |                       |
|--|-----------------------|
| No. Tarea: 20  | No. Historia: 8       |
| Nombre de la tarea: modificar secuencia de actividades.      |                       |
| Tipo de tarea: Desarrollo                                    | Puntos estimados: 0.1 |
| Fecha inicio: 10/02/2014                                     | Fecha fin: 10/02/2014 |
| Programador responsable: Deysi Columbié Hernández            |                       |
| Descripción: Permite modificar una secuencia de actividades. |                       |

**Tabla 3.26 Tarea de programación eliminar secuencia de actividades**

| Tarea de Programación                                       |                       |
|---|-----------------------|
| No. Tarea: 21   | No. Historia: 8       |
| Nombre de la tarea: eliminar secuencia de actividades.      |                       |
| Tipo de tarea: Desarrollo                                   | Puntos estimados: 0.1 |
| Fecha inicio: 11/02/2014                                    | Fecha fin: 11/02/2014 |
| Programador responsable: Deysi Columbié Hernández           |                       |
| Descripción: Permite eliminar una secuencia de actividades. |                       |

**Tabla 3.27 Tarea de programación insertar miembro de proyecto**

| Tarea de Programación |
|-----------------------|
|-----------------------|

|  |                       |
|--|-----------------------|
| No. Tarea: 22                                      | No. Historia: 8       |
| Nombre de la tarea: insertar miembro de proyecto.  |                       |
| Tipo de tarea: Desarrollo                          | Puntos estimados: 0.2 |
| Fecha inicio: 11/02/2014                           | Fecha fin: 12/02/2014 |
| Programador responsable: Deysi Columbié Hernández  |                       |
| Descripción: Permite insertar miembro de proyecto. |                       |

**Tabla 3.28 Tarea de programación mostrar miembro de proyecto**

|   |                       |
|---|-----------------------|
| <b>Tarea de Programación</b>                      |                       |
| No. Tarea: 23                                     | No. Historia: 8       |
| Nombre de la tarea: mostrar miembro de proyecto.  |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 13/02/2014                          | Fecha fin: 13/02/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite mostrar miembro de proyecto. |                       |

**Tabla 3.29 Tarea de programación Modificar miembro de proyectos**

|   |                       |
|---|-----------------------|
| <b>Tarea de Programación</b>                        |                       |
| No. Tarea: 24                                       | No. Historia: 9       |
| Nombre de la tarea: modificar miembro de proyecto   |                       |
| Tipo de tarea: Desarrollo                           | Puntos estimados: 0.2 |
| Fecha inicio: 14/02/2014                            | Fecha fin: 15/02/2014 |
| Programador responsable: Deysi Columbié Hernández   |                       |
| Descripción: Permite modificar miembro de proyecto. |                       |

**Tabla 3.30 Tarea de programación Eliminar miembro de proyectos**

| Tarea de Programación                              |                       |
|--|-----------------------|
| No. Tarea: 25                                      | No. Historia: 10      |
| Nombre de la tarea: eliminar miembro de proyecto.  |                       |
| Tipo de tarea: Desarrollo                          | Puntos estimados: 0.1 |
| Fecha inicio: 20/03/2014                           | Fecha fin: 20/03/2014 |
| Programador responsable: Deysi Columbié Hernández  |                       |
| Descripción: Permite eliminar miembro de proyecto. |                       |

**Tabla 3.31 Tarea de programación Insertar documentos**

| Tarea de Programación                             |                       |
|---|-----------------------|
| No. Tarea: 26                                     | No. Historia: 10      |
| Nombre de la tarea: insertar documentos.          |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 21/03/2014                          | Fecha fin: 21/03/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite insertar documentos.         |                       |

**Tabla 3.32 Tarea de programación Mostrar documentos**

| Tarea de Programación                             |                       |
|---|-----------------------|
| No. Tarea: 27                                     | No. Historia: 10      |
| Nombre de la tarea: mostrar documento.            |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.2 |
| Fecha inicio: 22/03/2014                          | Fecha fin: 23/03/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |

Descripción: Permite mostrar los datos del documento.

**Tabla 3.33 Tarea de programación Modificar documento**

| Tarea de Programación                             |                       |
|---|-----------------------|
| No. Tarea: 28                                     | No. Historia: 10      |
| Nombre de la tarea: modificar documento.          |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 24/03/2014                          | Fecha fin: 24/03/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite modificar documento.         |                       |

**Tabla 3.34 Tarea de programación Eliminar documento**

| Tarea de Programación                             |                       |
|---|-----------------------|
| No. Tarea: 29                                     | No. Historia: 11      |
| Nombre de la tarea: eliminar documento.           |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 25/03/2014                          | Fecha fin: 25/03/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite eliminar el documento        |                       |

**Tabla 3.35 Tarea de programación Insertar plan de resultado**

| Tarea de Programación                          |                  |
|--|------------------|
| No. Tarea: 30                                  | No. Historia: 12 |
| Nombre de la tarea: insertar plan de resultado |                  |

|   |                       |
|---|-----------------------|
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 26/03/2014                          | Fecha fin: 26/03/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite insertar plan de resultado   |                       |

**Tabla 3.36 Tarea de programación Mostrar plan de resultado**

|   |                       |
|---|-----------------------|
| <b>Tarea de Programación</b>                      |                       |
| No. Tarea: 31                                     | No. Historia: 13      |
| Nombre de la tarea: mostrar plan de resultado.    |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 27/03/2014                          | Fecha fin: 27/03/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite mostrar el plan de resultado |                       |

**Tabla 3.37 Tarea de programación Modificar plan de resultado**

|   |                       |
|---|-----------------------|
| <b>Tarea de Programación</b>                      |                       |
| No. Tarea: 32                                     | No. Historia: 14      |
| Nombre de la tarea: modificar plan de resultado.  |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 29/03/2014                          | Fecha fin: 29/03/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite modificar plan de resultado. |                       |

**Tabla 3.38 Tarea de programación Eliminar plan de resultado**



| Tarea de Programación                             |                       |
|---|-----------------------|
| No. Tarea: 33                                     | No. Historia: 15      |
| Nombre de la tarea: eliminar plan de resultado.   |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 30/03/2014                          | Fecha fin: 30/03/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite eliminar plan de resultado   |                       |

**Tabla 3.39 Tarea de programación Registrar software**

| Tarea de Programación                                 |                       |
|---|-----------------------|
| No. Tarea: 34   | No. Historia: 16      |
| Nombre de la tarea: registrar software.               |                       |
| Tipo de tarea: Desarrollo                             | Puntos estimados: 0.1 |
| Fecha inicio: 29/03/2014                              | Fecha fin: 29/03/2014 |
| Programador responsable: Deysi Columbié Hernández     |                       |
| Descripción: Permite registrar los datos del software |                       |

**Tabla 3.40 Tarea de programación Listar registro de software**

| Tarea de Programación   |                       |
|---|-----------------------|
| No. Tarea: 35   | No. Historia: 17      |
| Nombre de la tarea: listar registro de software.                      |                       |
| Tipo de tarea: Desarrollo   | Puntos estimados: 0.1 |
| Fecha inicio: 01/04/2014  | Fecha fin: 01/04/2014 |
| Programador responsable: Deysi Columbié Hernández                     |                       |
| Descripción: Permite mostrar un listado de los registros de software. |                       |

**Tabla 3.41 Tarea de programación Insertar artículo**

| Tarea de Programación                             |                       |
|---|-----------------------|
| No. Tarea: 36                                     | No. Historia: 18      |
| Nombre de la tarea: insertar artículo.            |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 02/04/2014                          | Fecha fin: 02/04/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite insertar artículo.           |                       |

**Tabla 3.41 Tarea de programación Mostrar un artículo**

| Tarea de Programación                             |                       |
|---|-----------------------|
| No. Tarea: 37                                     | No. Historia: 19      |
| Nombre de la tarea: Mostrar un artículo.          |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 03/04/2014                          | Fecha fin: 03/04/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite Mostrar un artículo          |                       |

**Tabla 3.42 Tarea de programación Modificar un artículo**

| Tarea de Programación                     |                       |
|---|-----------------------|
| No. Tarea: 38                             | No. Historia: 20      |
| Nombre de la tarea: modificar un artículo |                       |
| Tipo de tarea: Desarrollo                 | Puntos estimados: 0.1 |

|   |                       |
|---|-----------------------|
| Fecha inicio: 03/04/2014                          | Fecha fin: 03/04/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite modificar un artículo        |                       |

**Tabla 3.43 Tarea de programación Eliminar artículo**

| Tarea de Programación                             |                       |
|---|-----------------------|
| No. Tarea: 39                                     | No. Historia: 21      |
| Nombre de la tarea: eliminar artículo.            |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 04/04/2014                          | Fecha fin: 04/04/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite eliminar un artículo         |                       |

**Tabla 3.44 Tarea de programación Insertar evento**

| Tarea de Programación                             |                       |
|---|-----------------------|
| No. Tarea: 40                                     | No. Historia: 22      |
| Nombre de la tarea: insertar evento.              |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 05/04/2014                          | Fecha fin: 05/04/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite insertar evento              |                       |

**Tabla 3.45 Tarea de programación Mostrar evento**

| Tarea de Programación                               |                       |
|---|-----------------------|
| No. Tarea: 41                                       | No. Historia: 23      |
| Nombre de la tarea: mostrar evento.                 |                       |
| Tipo de tarea: Desarrollo                           | Puntos estimados: 0.1 |
| Fecha inicio: 06/04/2014                            | Fecha fin: 06/04/2014 |
| Programador responsable: Deysi Columbié Hernández   |                       |
| Descripción: Permite mostrar los datos de un evento |                       |

**Tabla 3.46 Tarea de programación Modificar evento**

| Tarea de Programación                             |                       |
|---|-----------------------|
| No. Tarea: 42                                     | No. Historia: 24      |
| Nombre de la tarea: modificar evento.             |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 07/04/2014                          | Fecha fin: 07/04/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite modificar un evento          |                       |

**Tabla 3.47 Tarea de programación Eliminar evento**

| Tarea de Programación                             |                       |
|---|-----------------------|
| No. Tarea: 43                                     | No. Historia: 24      |
| Nombre de la tarea: eliminar evento.              |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 08/04/2014                          | Fecha fin: 08/04/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |

Descripción: Permite eliminar un evento

**Tabla 3.48 Tarea de programación Insertar premio**

| Tarea de Programación                             |                       |
|---|-----------------------|
| No. Tarea: 44                                     | No. Historia: 24      |
| Nombre de la tarea: insertar premio.              |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.2 |
| Fecha inicio: 09/04/2014                          | Fecha fin: 10/04/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite insertar premio              |                       |

**Tabla 3.49 Tarea de programación Mostrar premio**

| Tarea de Programación                                |                       |
|--|-----------------------|
| No. Tarea: 45  | No. Historia: 24      |
| Nombre de la tarea: mostrar premio.                  |                       |
| Tipo de tarea: Desarrollo                            | Puntos estimados: 0.1 |
| Fecha inicio: 11/04/2014                             | Fecha fin: 11/04/2014 |
| Programador responsable: Deysi Columbié Hernández    |                       |
| Descripción: Permite mostrar los datos de un premio. |                       |

**Tabla 3.50 Tarea de programación Modificar premio**

| Tarea de Programación                 |                  |
|---------------------------------------|------------------|
| No. Tarea: 46                         | No. Historia: 25 |
| Nombre de la tarea: modificar premio. |                  |

|   |                       |
|---|-----------------------|
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 12/04/2014                          | Fecha fin: 12/04/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite modificar un premio          |                       |

**Tabla 3.51 Tarea de programación Eliminar premio**

|   |                       |
|---|-----------------------|
| <b>Tarea de Programación</b>                      |                       |
| No. Tarea: 47                                     | No. Historia: 26      |
| Nombre de la tarea: Eliminar premio.              |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 13/04/2014                          | Fecha fin: 13/04/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite eliminar un premio.          |                       |

**Tabla 3.52 Tarea de programación Registrar Publicación**

|   |                       |
|---|-----------------------|
| <b>Tarea de Programación</b>                      |                       |
| No. Tarea: 48                                     | No. Historia: 27      |
| Nombre de la tarea: Registrar publicación.        |                       |
| Tipo de tarea: Desarrollo                         | Puntos estimados: 0.1 |
| Fecha inicio: 14/04/2014                          | Fecha fin: 14/04/2014 |
| Programador responsable: Deysi Columbié Hernández |                       |
| Descripción: Permite registrar publicación        |                       |

**Tabla 3.53 Tarea de programación Listar registro de publicaciones**

| Tarea de Programación  |                       |
|--|-----------------------|
| No. Tarea: 49  | No. Historia: 28      |
| Nombre de la tarea: Listar registro de publicaciones                       |                       |
| Tipo de tarea: Desarrollo  | Puntos estimados: 0.1 |
| Fecha inicio: 15/04/2014   | Fecha fin: 15/04/2014 |
| Programador responsable: Deysi Columbié Hernández                          |                       |
| Descripción: Permite mostrar un listado de los registros de publicaciones. |                       |

## Anexo 4 Pruebas de Aceptación

Tabla 3.54 Prueba para comprobar la entrada de un usuario al sistema

| Prueba de aceptación  |
|---|
| Historia de usuario # 1: Autenticar usuarios  |
| Nombre: Prueba para comprobar la entrada de un usuario al sistema.  |
| Descripción: Validación de entrada de los datos del usuario.  |
| Condiciones de ejecución: El usuario debe introducir su nombre y contraseña.  |
| Entrada/Pasos de ejecución: El usuario escribe sus datos y pulsa "Entrar".  |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si el usuario tiene acceso para entrar a la aplicación e inserta sus datos correctamente entrara sin problemas al sistema.</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>➤ Se dé clic en el botón Entrar sin insertar nada en los campos de texto.</li></ul> |
| Evaluación de la prueba: Aceptada.  |

Tabla 3.55 Prueba para comprobar la entrada de datos de un usuario

| Prueba de aceptación   |
|--|
| Historia de usuario # 2: Gestionar usuarios                    |
| Nombre: Prueba para comprobar la entrada de datos del usuario. |
| Descripción: Validación de entrada de los datos del usuario.   |



|   |
|---|
| Condiciones de ejecución: El usuario debe introducir los datos de un usuario.   |
| Entrada/Pasos de ejecución: El usuario escribe los datos del usuario.   |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si se introducen los datos correctamente se creará el nuevo usuario.</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>➤ Se inserten los datos de un usuario ya creado.</li><li>➤ Se inserte en el campo Email un correo no válido.</li><li>➤ Se dejen campos obligatorios vacíos.</li></ul> |
| Evaluación de la prueba: Aceptada.  |

**Tabla 5.56: Prueba para comprobar la entrada de datos de la línea de investigación.**

|  |
|--|
| Prueba de aceptación   |
| Historia de usuario # 3: Gestionar línea de investigación  |
| Nombre: Prueba para comprobar la entrada de datos de la línea de investigación.  |
| Descripción: Validación de entrada de los datos de la línea de investigación.  |
| Condiciones de ejecución: El usuario debe introducir los datos de la línea de investigación.   |
| Entrada/Pasos de ejecución: El usuario escribe los datos de la línea de investigación.   |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si se introducen los datos correctamente se insertará la nueva línea de investigación.</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>➤ Se dejen campos obligatorios vacíos.</li></ul> |
| Evaluación de la prueba: Aceptada.   |

**Tabla 3.57 Prueba para comprobar la entrada de datos de indicadores dinámicos de evaluación.**

|   |
|---|
| Prueba de aceptación  |
| Historia de usuario # 4: Gestionar indicadores dinámicos de evaluación  |
| Nombre: Prueba para comprobar la entrada de datos de los indicadores dinámicos.   |
| Descripción: Validación de entrada de los datos de los indicadores.   |
| Condiciones de ejecución: El usuario debe introducir los datos del indicador.   |
| Entrada/Pasos de ejecución: El usuario escribe los datos del indicador dinámico de evaluación.  |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si se introducen los datos correctamente se insertará el indicador.</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>➤ Se dejen campos obligatorios vacíos.</li></ul> |
| Evaluación de la prueba: Aceptada.  |

**Tabla 3.58: Prueba para comprobar la entrada de datos del proyecto**

|   |
|---|
| Prueba de aceptación  |
| Historia de usuario # 5: Gestionar proyecto.  |
| Nombre: Prueba para comprobar la entrada de datos del proyecto.   |
| Descripción: Validación de entrada de los datos del proyecto.   |
| Condiciones de ejecución: El usuario debe introducir los datos del proyecto   |
| Entrada/Pasos de ejecución: El usuario escribe los datos que se pide para insertar el proyecto.   |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si se insertan los datos correctamente se insertará el proyecto</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>➤ Se dejen campos obligatorios vacíos.</li></ul> |
| Evaluación de la prueba: Aceptada.  |



**Tabla 3.59: Prueba para comprobar la entrada de datos la secuencia de actividades**

|  |
|--|
| Prueba de aceptación   |
| Historia de usuario # 6: Gestionar secuencia de actividades  |
| Nombre: Prueba para comprobar la entrada de datos la secuencia de actividades.   |
| Descripción: Validación de entrada de los datos de la secuencia de actividades.  |
| Condiciones de ejecución: El usuario debe introducir los datos de la secuencia de actividades.   |
| Entrada/Pasos de ejecución: El usuario escribe los datos de la secuencia de actividades.   |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si se insertan los datos correctamente se creará un nuevo registro en la tabla secuencia de actividades.</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>➤ Se dejen campos obligatorios vacíos.</li></ul> |
| Evaluación de la prueba: Aceptada.   |

**Tabla 3.60: Prueba para comprobar la entrada de datos del miembro de proyecto**

|   |
|---|
| Prueba de aceptación  |
| Historia de usuario # 7: Gestionar miembro de proyecto  |
| Nombre: Prueba para comprobar la entrada de datos del miembro de proyecto   |
| Descripción: Validación de entrada de los datos del miembro de proyecto.  |
| Condiciones de ejecución: El usuario introduce los datos correspondientes.  |
| Entrada/Pasos de ejecución: El usuario escribe los datos del miembro de proyecto.   |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si se insertan los datos correctamente se inserta el miembro de proyecto.</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>➤ Se dejen campos obligatorios vacíos.</li></ul> |
| Evaluación de la prueba: Aceptada.  |



**Tabla 3.61: Prueba para comprobar la entrada de datos del documento**

|   |
|---|
| Prueba de aceptación  |
| Historia de usuario # 8: Gestionar documento  |
| Nombre: Prueba para comprobar la entrada de datos del documento   |
| Descripción: Validación de entrada de los datos del documento   |
| Condiciones de ejecución: El usuario introduce los datos correspondientes.  |
| Entrada/Pasos de ejecución: El usuario escribe los datos del documento.   |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si se insertan los datos correctamente se inserta el documento.</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>➤ Se dejen campos obligatorios vacíos.</li></ul> |
| Evaluación de la prueba: Aceptada.  |

**Tabla 3.62: Prueba para comprobar la entrada de datos del plan de resultado.**

|  |
|--|
| Prueba de aceptación   |
| Historia de usuario # 9: Gestionar plan de resultado   |
| Nombre: Prueba para comprobar la entrada de datos del plan de resultado  |
| Descripción: Validación de entrada de los datos del plan de resultado  |
| Condiciones de ejecución: El usuario introduce los datos correspondientes.   |
| Entrada/Pasos de ejecución: El usuario escribe los datos del plan de resultado.  |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si se insertan los datos correctamente se inserta el plan.</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>➤ Se dejen campos obligatorios vacíos.</li></ul> |
| Evaluación de la prueba: Aceptada.   |



**Tabla 3.63: Prueba para comprobar la entrada de datos de registro de software**

|  |
|--|
| Prueba de aceptación   |
| Historia de usuario # 10: Registrar software   |
| Nombre: Prueba para comprobar la entrada de datos del registro de software   |
| Descripción: Validación de entrada de los datos del registro de software.  |
| Condiciones de ejecución: El usuario introduce los datos correspondientes.   |
| Entrada/Pasos de ejecución: El usuario escribe los datos del registro de software  |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si se insertan los datos correctamente se inserta el registro de software.</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>➤ Se dejen campos obligatorios vacíos.</li></ul> |
| Evaluación de la prueba: Aceptada.   |

**Tabla 3.64: Prueba para comprobar la entrada de datos del artículo**

|  |
|--|
| Prueba de aceptación   |
| Historia de usuario # 11: Gestionar artículo   |
| Nombre: Prueba para comprobar la entrada de datos del artículo   |
| Descripción: Validación de entrada de los datos del artículo.  |
| Condiciones de ejecución: El usuario introduce los datos correspondientes.   |
| Entrada/Pasos de ejecución: El usuario escribe los datos del artículo  |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si se insertan los datos correctamente se inserta el artículo.</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>➤ Se dejen campos obligatorios vacíos.</li></ul> |
| Evaluación de la prueba: Aceptada.   |





**Tabla 3.65: Prueba para comprobar la entrada de datos del evento**

|   |
|---|
| Prueba de aceptación  |
| Historia de usuario # 12: Gestionar evento  |
| Nombre: Prueba para comprobar la entrada de datos del evento  |
| Descripción: Validación de entrada de los datos del evento.   |
| Condiciones de ejecución: El usuario introduce los datos correspondientes.  |
| Entrada/Pasos de ejecución: El usuario escribe los datos del evento   |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si se insertan los datos correctamente se inserta el evento</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>➤ Se dejen campos obligatorios vacíos.</li></ul> |
| Evaluación de la prueba: Aceptada.  |

**Tabla 3.66: Prueba para comprobar la entrada de datos del premio**

|  |
|--|
| Prueba de aceptación   |
| Historia de usuario # 13: Gestionar premio   |
| Nombre: Prueba para comprobar la entrada de datos del premio   |
| Descripción: Validación de entrada de los datos del premio.  |
| Condiciones de ejecución: El usuario introduce los datos correspondientes.   |
| Entrada/Pasos de ejecución: El usuario escribe los datos del premio  |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si se insertan los datos correctamente se inserta el premio.</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>➤ Se dejen campos obligatorios vacíos.</li></ul> |
| Evaluación de la prueba: Aceptada.   |

**Tabla 6.67: Prueba para comprobar la entrada de datos del registro de publicación**

|   |
|---|
| Prueba de aceptación  |
| Historia de usuario # 14: Gestionar registro de publicación   |
| Nombre: Prueba para comprobar la entrada de datos del registro de publicación   |
| Descripción: Validación de entrada de los datos del registro de publicación.  |
| Condiciones de ejecución: El usuario introduce los datos correspondientes.  |
| Entrada/Pasos de ejecución: El usuario escribe los datos del registro de publicación  |
| Resultado esperado: <ul style="list-style-type: none"><li>➤ Si se insertan los datos correctamente se inserta el registro de publicación.</li></ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>➤ Se dejen campos obligatorios vacíos.</li></ul> |
| Evaluación de la prueba: Aceptada.  |

## Anexo 5 Interfaz de Usuario

Vicerrectoría de Investigación y Postgrado

The screenshot shows a web interface for adding a research project. The title is 'Proyecto de Investigación'. The form contains the following fields:

- Nombre del proyecto:** A text input field with the placeholder 'nombre del proyecto'.
- Estado del proyecto:** A text input field with the placeholder 'estado del proyecto'.
- Fecha inicio:** A date picker field with the placeholder 'fecha de inicio' and a calendar icon.
- Fecha de Finalización:** A date picker field with the placeholder 'fecha de finalización' and a calendar icon.
- Financiamiento:** A dropdown menu with the placeholder 'financiamiento' and a downward arrow icon.
- Línea de investigación:** A text input field with the placeholder 'línea de investigación'.
- Plan de indicadores:** A text input field with the placeholder 'plan de indicadores'.

At the bottom left of the form is a blue button with a white plus sign and the text '+ Insertar'.

**Figura 6** Interfaz de usuario Insertar proyecto de investigación

Vicerrectoría de Investigación y Postgrado

ISMMM

### Insertar Documento

Direccion

+ Insertar

**Figura 7** Interfaz de usuario insertar documento

Vicerrectoría de Investigación y Postgrado

ISMMM

### Insertar Usuario

Nombre Usuario: nombre usuario

Nombre Completo: nombre completo

Password: password

Repetir password: repetir password

Email: email

Nivel Acceso: --- Seleccione ---

+ Insertar

**Figura 8** Interfaz de usuario Insertar usuario

Vicerrectoría de Investigación y Postgrado

ISMMM

### Insertar Miembro de Proyecto

Nombre

Apellidos

Proyecto de investigación

**Figura 9** Interfaz de usuario Insertar miembro de proyecto.