Universidad de Matanzas

Facultad de Ciencias Técnicas

Especialidad Ingeniería Informática

Trabajo para optar por el Título de Ingeniero en Informática.

"Sistema informático en apoyo al desarrollo del balance de ciencia y técnica en la Universidad de Matanzas"

Autor: Jorge Martínez Pérez

Tutor: MSc. Mayli Estopiñán Lantigua

MSc. Yadrián A. García Pulido

Matanzas, 2018

Pensamientos

"Así como perfeccionamos las ciencias, debemos perfeccionar la moralidad, sin la cual el saber se destruye"

J. NEWTON

"El mayor enemigo del conocimiento no es la ignorancia, sino la ilusión del conocimiento"

Stephen Hawking

Dedicatoria

. A mis padres y familia por todo el apoyo que siempre me han dado.

. A mi novia por estar siempre conmigo.

. A mis amigos por su apoyo y preocupación en esta etapa tan difícil.

Agradecimientos

. A mis padres porque gracias a ellos he llegado hasta aquí.

. A todos los profesores que de una forma u otra me ayudaron con el trabajo en especial a Omar Rivero, Maily Estupinan, Yadrian García.

Opinión del Tutor del Trabajo de Diploma

Resumen

El presente Trabajo de Diploma fue realizado en la Universidad de Matanzas, ubicada en la provincia de Matanzas, municipio Matanzas, se fundamenta en diseñar un sistema que permita facilitar la realización de la parte cuantitativa del balance de ciencia y técnica. Entre las facilidades que brinda el sistema se encuentran la generación automática de reportes y estadísticas útiles para tomar decisiones posteriores sobre la realización del balance de ciencia y técnica; así como la gestión de las diferentes facultades, carreras, bases de datos y grupos correspondientes a las publicaciones, así como los proyectos existentes y premios obtenidos. Se plantean los métodos teóricos y empíricos que se utilizaron en la investigación. Se desarrolló una aplicación web de gestión que automatiza el proceso de la parte cuantitativa del balance de ciencia y técnica de la Universidad de Matanzas lo cual permite agilizar muchos procesos que se realizan actualmente de forma indirecta y casi manual.

Summary

This Diploma Work was carried out at the University of Matanzas, located in the province of Matanzas, Matanzas municipality. It is based on designing a system that facilitates the realization of the quantitative part of the science and technology balance. Among the facilities provided by the system are the automatic generation of reports and useful statistics to make subsequent decisions on the realization of the balance of science and technology; as well as the management of the different faculties, careers, databases and groups corresponding to the publications, as well as the existing projects and obtained prizes. The theoretical and empirical methods that were used in the investigation are presented. A web management aplication was developed that automates the process of the quantitative part of the science and technology balance of the University of Matanzas, which makes it possible to streamline many processes that are currently carried out indirectly and almost manually.

Índice

Contenido

Índice de ilustraciones

Índice de tablas

Introducción

La **ciencia** (del latín *scientĭa* 'conocimiento') es el conjunto ordenado de conocimientos estructurados sistemáticamente. La ciencia es el conocimiento que se obtiene 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 sistemas organizados por medio de un método científico (Wikipedia, 2018).

El desarrollo de la ciencia es un proceso muy complejo. Sus progresos son realmente fascinantes, debido a sus efectos sobre la marcha del mundo y la vida de las personas, que requieren reflexión crítica. Esta reflexión está relacionada con el hecho de que hoy la ciencia es, de hecho, un activo imprescindible de una sociedad moderna, una cuestión pública de creciente importancia.

Aunque la percepción que tiene un científico de su actividad sea intensamente personal y aunque pueda ser con toda sinceridad ajena a otros aspectos de su quehacer, sería un error si concibiese la ciencia simplemente como el objeto de su interés. Por el contrario, ésta es una actividad que tiene lugar en el seno de la sociedad y, en términos generales, está sujeta a los normales condicionamientos culturales, ideológicos, políticos y económicos. Por ello, la percepción pública de la ciencia y la inserción de la misma en el mundo siempre van cambiando en consonancia con los cambios sociales a través de la historia, aunque no todos estos procesos son de la misma envergadura.

La publicación de artículos es una acción que reviste gran importancia en el ámbito científico. Es el fruto del trabajo de investigadores que se forman principalmente en los Programas Doctorales en las universidades e institutos de investigación quienes son capaces de producir conocimientos especializados en diferentes áreas mediante la concepción, la realización y la defensa pública de

una tesis que constituya un aporte relevante a la ciencia, la tecnología o las humanidades.

La producción científica de las universidades y concretamente los artículos de difusión internacional son, a día de hoy, el valor con el que se mide la calidad de estas instituciones en lo que a capacidad investigadora se refiere. Es por ello que el principal objeto de estudio de bibliómetras y documentalistas dentro del campo de la investigación universitaria atañe habitualmente a la recolección de datos de esta índole y su posterior incorporación a los sistemas de gestión de la información de las universidades. No obstante, hay una parte previa a la publicación del documento con los resultados científicos que habitualmente es obviada por los técnicos de la información. Esta parte es la que hace referencia a la obtención de fondos para llevar a cabo dicha producción, así como la gestión de los mismos durante su ciclo de vida. Este proceso, sin el cual la obtención de conocimientos científicos seria, si bien no imposible, sí mucho más complicada, se conoce como Gestión de los Recursos de Investigación

En los últimos cuatro años se han desarrollado notablemente los sistemas automatizados de gestión editorial, se han empezado a implantar en publicaciones científicas en el mundo y la oferta de programas ha ido creciendo. Algunos de ellos se han consolidado y han demostrado sus ventajas para agilizar el trabajo de edición y mejorar algunos aspectos de la calidad de las revistas.

Después del triunfo de la Revolución el quehacer científico experimentó un avance vertiginoso, convirtiéndose en uno de los pilares del desarrollo del país, bajo la premisa formulada por el líder de la Revolución Fidel Castro cuando expresó: “El futuro de nuestra patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento”

En Cuba el CITMA tiene la misión de dirigir, ejecutar y controlar la política del estado y del gobierno en materia de ciencia, tecnología, medio ambiente; el uso de la energía nuclear, de normalización, metrología y control de la calidad, propiciando la integración coherente de estas para contribuir al desarrollo sostenible del país.

En la Universidad de Matanzas se realiza anualmente un balance de ciencia y técnica el cual precisa los principales resultados de la ciencia, técnica e

investigación obtenidos en el año que hayan sido avalados como logros por el Consejo Científico universitario además de los principales impactos económicos, sociales y de otro tipo, que fueron obtenidos en las prioridades nacionalmente establecidas, como consecuencia de la aplicación y generalización de los resultados de investigación obtenidos por la Universidad de Matanzas. Los impactos descritos son resultados ya aplicados y deben haber sido aprobados como impactos por el Consejo Científico universitario.

De lo que se deriva el siguiente ***Problema de la investigación:***

¿Cómo facilitar el proceso para realizar el balance de ciencia y técnica de la Universidad de Matanzas?

El **objeto de estudio** es la gestión del balance de ciencia técnica de la Universidad de Matanzas.

Por ellos se propone el siguiente ***objetivo general***: desarrollar un sistema de gestión que facilite el proceso de realizar el balance de ciencia técnica de la UMCC Camilo Cienfuegos.

El **campo de acción** es la automatización del balance de ciencia técnica de la Universidad de Matanzas.

Para resolver este problema se plantea la siguiente ***Hipótesis:***

Si se desarrolla un sistema de gestión entonces se facilitará el proceso de realizar el balance de ciencia y técnica de la Universidad de Matanzas.

Los ***objetivos específicos*** son:

- ✓ Analizar el marco teórico referencial del balance de ciencia y técnica de la UMCC Camilo Cienfuegos.
- ✓ Emplear una metodología de desarrollo de software para la aplicación web.
- ✓ Utilizar los lenguajes de programación adecuados para el desarrollo de la aplicación web.
- ✓ Validar la puesta en marcha del sistema, por medio de las pruebas que propone la metodología de desarrollo de software y el cumplimiento de los requerimientos funcionales.

Para desarrollar el software se emplean diversos métodos los cuales se dividen en métodos teóricos: histórico-lógico, analítico-sintético, inductivo-deductivo, hipotético-deductivo y en métodos empíricos: observación entrevistas y análisis de documentos.

Se utiliza una metodología de desarrollo de software ágil, Programación Extrema (XP), que permite crear Historias de Usuarios (HU) en colaboración con el cliente.

Como beneficios se pueden señalar el progreso en las condiciones de trabajo, en el proceso de reservación de servicios, alojamientos y la renta de autos. Además, agilizar las ofertas lo cual repercute de forma positiva en los clientes.

Se facilita una herramienta de gran aporte práctico porque permite una automatización del balance de ciencia y técnica de la Universidad de Matanzas.

El documento se conforma en tres capítulos. El primero constituye el Marco Teórico-Referencial, donde se plantean las principales definiciones relacionadas con el campo de acción, se exponen las características fundamentales de los lenguajes de programación los sistemas de base de datos y la metodología empleada.

El capítulo dos se denomina Análisis, Diseño de la Solución Propuesta y Construcción, el mismo describe la solución propuesta al problema de investigación. Presenta una planificación inicial del proyecto usando la metodología XP, y desarrolla la propuesta para darle solución a la situación problemática mediante una planificación por iteraciones.

El capítulo tres es la Validación de la Solución Propuesta. Se realizan pruebas funcionales y un estudio de los beneficios tangibles e intangibles. Se obtienen resultados basados en el criterio del cliente y la metodología empleada.

Posteriormente las Conclusiones que permiten arribar a resultados específicos que son obtenidos mediante la realización del software y del documento. Las Recomendaciones sirven como punto de partida para nuevas investigaciones relacionadas con la temática abordada.

Se presentan al final del documento las Bibliografías empleadas y los Anexos que sirven de apoyo al trabajo.

Capítulo I: Marco teórico-referencial

En el presente capítulo se realiza un estudio sobre el desarrollo y las perspectivas del balance de ciencia y técnica de la Universidad de Matanzas. Se analiza la información referida al tema para establecer una estructura y un flujo de procesos para realizar dicho balance. Se considera la opción de la creación de una aplicación web para gestionar los mismos. Se hace mención a los diferentes métodos que se emplean. Se explica la metodología Programación Extrema y las tecnologías usadas en el desarrollo del sistema. Para desarrollar estos temas se realiza una búsqueda bibliográfica que permite consolidar las bases teóricas para favorecer la comprensión de contenidos posteriores y la resolución del problema que se plantea.

1.1 Fundamento y antecedentes del trabajo

Para la gestión del proceso de realización de la parte cuantitativa del balance de ciencia y técnica de la Universidad de Matanzas cuenta con un modelo llamado Anexo1 de la plataforma sistema informático de entorno de escritorio desarrollado sobre la plataforma Access de Microsoft Office. Actualmente cada jefe de departamento debe entregar el trabajo científico de cada uno de sus miembros a la encargada de la realización del balance la cual asegura que se entrega hasta escrito manualmente lo cual no ocurre en todos los casos, la falta de uniformidad en este proceso crea un cúmulo de dificultades que impiden una rápida y correcta realización.

También como antecedente tenemos el Sistema Automatizado para la Gestión de la Información Ciencia y Técnica de un Departamento Docente Universitario que es un software que permite la automatización del proceso de gestión de la información referida a las publicaciones realizadas por los profesores de un departamento docente, su participación en eventos, proyectos y servicios científicos, así como los premios obtenidos, este software constituye una gran base para la implementación de la aplicación web que se propone en este documento el cual pasa de la escala de los departamentos vistos individualmente a verlos como un conjunto en la universidad, este nuevo software incorpora la gestión de las diferentes facultades, carreras, grupos, bases de datos, tipos de proyectos, grados científicos, categorías docentes y tipos de premios para

impedir futuros mantenimiento del software debido al cambio de los factores mencionados anteriormente.

1.2 Objeto de estudio

1.2.1 Caracterización de la Universidad de Matanzas

La entidad Universidad de Matanzas, ubicada en Autopista a Varadero km 3, provincia Matanzas, Cuba, es una destacada institución universitaria que ha experimentado desde sus inicios un vertiginoso y potente desarrollo educacional, llegando a ser reconocida una de las cinco mejores universidades cubanas, según el Ministerio de Educación Superior (MES). Sus orígenes se remontan al 9 de mayo de 1972, cuando con apoyo de la Universidad de La Habana y del Comité Provincial del Partido Comunista de Cuba se creó la Sede Universitaria de Matanzas.

El objeto de estudio de esta propuesta consiste en mejorar la realización del balance de ciencia y técnica, haciendo uso de las últimas herramientas tecnológicas disponibles para la web, y de este modo satisfacer las exigencias del cliente. Representa una fuente automática y de recepción que funciona de manera continua, permitiendo respuestas rápidas lo cual mejora la calidad y eficiencia.

1.2.2 Descripción general de la Universidad de Matanzas

La Universidad de Matanzas hoy es muestra de cuánto ha avanzado la Educación Superior en la provincia, al tener alrededor de 32 000 estudiantes (23 000 de pregrado y 9 000 de postgrado) y un claustro de más de 300 profesores. Posee publicaciones como: Periódico "El Universitario", Revista Cubana de Investigaciones Turísticas Retos Turísticos (ISSN 1681_9713), Revista Pastos y Forrajes (ISSN 0864_0394), Revista Científica Estudiantil de Investigaciones Turísticas "Investur". Entre los principales méritos obtenidos por la UMCC a lo largo de sus 38 años de trayectoria y como resultado del esfuerzo y empeño demostrado en investigaciones que han alcanzado un alto impacto científico, económico y social, se encuentran:

- 17 premios de la Academia de Ciencias de Cuba.

- Orden "Carlos J. Finlay", otorgada por el Consejo de Estado de la República de Cuba a la EEPFIH.
- 4 premios nacionales del Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMA).
- 13 patentes, de las cuales 2 son internacionales.
- Más de 150 publicaciones científicas en revistas de alta visibilidad del *Science Citations Index*.

Su misión es garantizar la formación integral revolucionaria de profesionales de las ramas de Ciencias Técnicas, Económicas, Agropecuarias y Humanísticas, contribuyendo a la superación continua de graduados universitarios y cuadros, desarrollando la ciencia e innovación tecnológica y la extensión universitaria y la informatización para el avance de la sociedad matancera y cubana.

1.2.3 Flujo del proceso involucrado en el campo de acción

En la Universidad de Matanzas para la realización del balance de ciencia y técnica los profesores informan a su respectivo jefe de departamento sobre sus aportes a la ciencia y técnica, el jefe de departamento recoge los siguientes datos nombre y apellidos del profesor, categoría docente, grado científico, nombre de la publicación, si está vinculada a un proyecto, si obtuvo algún premio o reconocimiento, el tipo de publicación: en caso de ser un libro o una monografía se recogen los siguientes datos el título, editora, código ISBN y el país; si es un artículo se recogerá el nombre del artículo, nombre de la revista en que fue publicado, fecha de publicación, volumen de la revista, país de la revista y base de datos donde se referencia la revista (al menos una, la más importante). El jefe de departamento entrega al encargado de realizar el balance de ciencia y técnica todos estos datos.

1.2.4 Análisis comparativo de propuestas existentes con la propuesta del trabajo

- Sistema de Gestión de la Investigación (SGI) de la Universidad de Talca, Chile (Iván F, 2007).

El sistema posee una arquitectura basada en roles (ejecutivos, investigadores, visitantes y otros) lo que ofrece una variada información y servicios atendiendo a su rol o perfil. Respecto a las investigaciones existentes, el sistema ofrece un conjunto de indicadores de gestión y estadísticas que dan al traste con el cumplimiento de los objetivos propuestos. La aplicación se encuentra estructurada atendiendo a los procesos por lo que atraviesa un proyecto de investigación y los procesos administrativos que los regulan. Dentro de las secciones que ofrece el sistema resalta el directorio de investigadores, centros tecnológicos, programas de investigación, proyectos adjudicados, publicaciones, eventos, documentos, estadísticas, sitios de interés y noticias. La metodología de desarrollo que se utilizó para la ejecución del proyecto fue RUP (del inglés Rational Unified Process), de la que se tomaron sus mejores prácticas.

- Sistema Integrado de Información sobre Investigación Científica y Tecnológica de México.

La estructura del sistema permite conjuntar los esfuerzos de las diferentes instituciones de educación superior, centros, organismos, empresas y personas, para incorporar toda la información sobre investigación científica y tecnológica, datos sobre técnicas y servicios que ofertan y demandan los participantes en actividades científicas y tecnológicas. Las bases de datos sobre participantes en programas y registros de Ciencia y Tecnología, así como sobre sus capacidades y producción se encuentran actualizadas, normalizadas y relacionadas. El SIICYT es un instrumento que apoya la divulgación de la ciencia y la tecnología y se logra una conciencia creciente en la sociedad sobre la importancia de la investigación y el conocimiento en la elevación del nivel de vida.

Los software extranjeros antes mencionados a pesar de tener una amplia utilidad, no cumplen con los requisitos necesarios para ser empleados en Cuba, estos son software privativos por lo que no existe una forma libre de acceso a su código fuente, el cual solo se encuentra a disposición de su desarrollador y no se permite su libre modificación, adaptación o incluso lectura por parte de terceros. El sistema que se proyecta diseñar, a pesar de ser menos complejo en cuanto a las prestaciones que sus competencias

extranjeras ofrecen, cumple con todos los requisitos que exige el cliente y se adecua a las condiciones.

(SIICYT, Marzo 2014.)

URL: http://www.siicyt.gob.mx/siicyt/quienesSomos.do?pSel=%27%27

1.3 Metodología de la investigación empleada

La metodología de la investigación es una disciplina de conocimiento encargada de elaborar, definir y sistematizar el conjunto de técnicas, métodos y procedimientos que se deben seguir durante el desarrollo de un proceso de investigación para la producción de conocimiento. Orienta la manera en que se enfoca una investigación y la forma en que se recolectan, analizan y clasifican los datos, con el objetivo de que los resultados tengan validez y pertinencia, y cumplan con los estándares de exigencia científica. La metodología de la investigación, en este sentido, es también la parte de un proyecto de investigación donde se exponen y describen razonadamente los criterios adoptados en la elección de la metodología, sea esta cuantitativa o cualitativa (Monguel, 2005).

1.3.1 Métodos teóricos empleados

Para el desarrollo del software se utilizaron diversos métodos tales como:

Histórico - Lógico: se emplea para realizar un análisis histórico de la realización del balance de ciencia y técnica en la Universidad de Matanzas, así como los fundamentos y variaciones que ha tenido el mismo durante el proceso de realización. Se analizan las ventajas y desventajas y se toman como punto de partida para lograr mejoras. De esta manera encontrar una forma lógica de desarrollar el sistema informático deseado.

Analítico - Sintético: se utiliza para los procesos de la información referida al control de las publicaciones. Mediante el mismo se logra un estudio detallado y con mayor envergadura de la bibliografía consultada, lo que permite aumentar la precisión de las bases teóricas, que reafirma la necesidad de la aplicación propuesta.

Inductivo - Deductivo: aportó una visión más amplia de la entidad y en que consiste su funcionamiento. Permitió establecer las principales ventajas de implementar una aplicación que pudiese ser empleada para facilitar la realización del balance de ciencia y técnica.

Hipotético-deductivo: es indispensable para confirmar lo que se supone en el inicio; además permite llegar a conclusiones y trazar posibles nuevas vías a partir de lo que se conocía.

1.3.2 Métodos empíricos empleados

Observación: mediante este proceso se pudo comprender el funcionamiento interno que posee la Universidad de Matanzas, así como la detección de los problemas.

Entrevistas: para lograr la comprensión del sistema de gestión y renta de autos fue necesario entrevistarse con la Jefa de departamento de CTI.

A través de ellas dieron respuestas a las principales inquietudes que permitieron la consolidación de los conocimientos necesarios para enfrentarse a la realización del sistema informático.

Análisis de documentos: permite la consulta de la bibliografía especializada, auxiliándose en los motores de búsqueda Google y Bing, lo que permite sentar las bases teóricas del trabajo. Además, se emplearon documentos en formato duro ofrecidos por la entidad que fueron de gran ayuda para el estudio.

1.4 Fundamentación de la metodología utilizada

El auge de la tecnología ha originado la necesidad de implantar Metodologías de Desarrollo de Software, que tienen el objetivo de agilizar y automatizar los procesos. Estas ayudan a entregar un producto de calidad en tiempo y costo estimados (Anexo 1). Las metodologías ágiles de desarrollo de software han despertado interés gracias a que proponen simplicidad y velocidad para crear sistemas (Highsmith, y otros, 2001). Las metodologías tradicionales no se adaptan a las nuevas necesidades o expectativas que tienen los usuarios, en parte porque los métodos usados no son flexibles ante la posibilidad de la exigencia de nuevos requerimientos (Cao, y otros, 2004). Estos cambios generalmente implican altos costos, demanda de tiempo y la reestructuración total del proyecto que se esté llevando; en contraparte, los métodos ágiles permiten un desarrollo iterativo y adaptable que permite la integración de nuevas funcionalidades a lo largo del desarrollo del proyecto; para que tanto el cliente como el desarrollador

queden satisfechos porque el producto final tiene una calidad adecuada (Giraldo, 2006).

El enfoque tradicional de desarrollo de aplicaciones heredado de otras ramas de la ingeniería, ha sido el de hacer un estudio exhaustivo del problema, establecer un plan y llevar a cabo la construcción. Es la metodología conocida como desarrollo en cascada, con las fases consecutivas de Análisis, Diseño, Codificación, Pruebas e Implementación.

El objetivo de las metodologías de desarrollo ágil de software es la organización de un trabajo creativo, que suele ser bastante caótico. Se intenta dar prioridad a la ejecución sobre la planificación. A medida que se profundiza en el conocimiento de un problema se cambian los planes. Cuando el cliente vea nuestras propuestas, se le ocurrirán nuevas ideas que cambiarán los planes. Cuando profundicemos en el conocimiento de nuevas tecnologías haremos descubrimientos que cambiarán de nuevo los planes.

Ágil quiere decir, adaptable. Desde esta premisa, los cambios son bienvenidos, derivan de un mejor entendimiento del problema y son una oportunidad para mejorar el software (Blanch, 2010).

El manifiesto ágil plantea que es necesario valorar más a los individuos y su interacción que a los procesos y las herramientas. Las herramientas mejoran la eficiencia, pero sin personas con conocimiento técnico y actitud adecuada, no producen resultado (Letelier, y otros, 2006). También afirma que en la mayoría de las ocasiones se crea el entorno y luego se espera a que el equipo se adapte a él, lo cual constituye un error que debe ser erradicado, ya que debe ser el equipo y no de forma contraria, el que en base a sus necesidades sea capaz de crear el entorno. Destaca la importancia de poder ver anticipadamente cómo se comportan las funcionalidades esperadas sobre prototipos o sobre las partes ya elaboradas del sistema final. El manifiesto ágil define al cliente como un miembro más del equipo, que se integra y colabora en el grupo de trabajo (José H, y otros, 2003).

Los principales valores de la gestión ágil son la anticipación y la adaptación; diferentes a los de la gestión de proyectos ortodoxa: planificación y control para evitar desviaciones sobre el plan (José H, y otros, 2003). Los principios son:

1. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
2. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
3. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas
4. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
5. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
6. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
7. El software que funciona es la medida principal de progreso.
8. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
10. La simplicidad es esencial.
11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
12. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

En cuanto a la superioridad de las metodologías no ágiles con respecto a las que lo son se considera que para lograr hacer frente a cualquier proyecto y se obtengan resultados favorables no se puede establecer una metodología única. Éstas deben ser adaptadas al contexto es decir los recursos humanos y técnicos, el tiempo de desarrollo, el tipo de sistema, es decir depende de lo que se desee lograr es necesario hacer un estudio para aplicar la metodología correcta.

1.5 Programación extrema

Programación extrema (*Extreme Programming*, XP) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software (Anexo 2), promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo (Joskowicz, 2008). XP 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 (Goto, y otros, 2014). XP se define como especialmente adecuada para proyectos con requisitos imprecisos donde existe un alto riesgo técnico (Beck, 2000).

Se muestran las características esenciales de XP organizadas en los apartados siguientes: historias de usuario, roles, proceso y prácticas.

1.5.1 Las Historias de Usuario

Las historias de usuario son descripciones cortas y simples de una funcionalidad, escritas desde la perspectiva de la persona que necesita una nueva capacidad de un sistema, por lo general el usuario, área de negocio o cliente.

Es la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (Jeffries,R, y otros, 2001).

1.5.2 Roles XP

Los roles de acuerdo con la propuesta original son:

Programador: escribe las pruebas unitarias y produce el código del sistema. Responsable de decisiones técnicas, de construir el sistema, sin distinción entre analistas, diseñadores o codificadores (Hurtado, y otros, 2005).

Cliente: escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de

usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio (Calero, 2003).

Encargado de pruebas *(Tester):* ayuda al cliente a escribir las pruebas funcionales. (Reynoso, 2012). Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas (Beck, 2000).

Encargado de seguimiento (*Tracker*): proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración (Cubel, y otros, 2012).

Entrenador (*Coach*): es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

Consultor: es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.

Gestor (*Big boss*): es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

1.5.3 Proceso XP

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

- ✓ El cliente define el valor de negocio a implementar.
- ✓ El programador estima el esfuerzo necesario para su implementación.
- ✓ El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- ✓ El programador construye ese valor de negocio.
- ✓ Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se pierde calidad en el software o no se cumplen los plazos establecidos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto (Beck, 2000).

1.5.4 Prácticas XP

La principal suposición que se realiza en XP, es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes prácticas (Wake, 2000).

- ✓ El juego de la planificación. Hay una comunicación frecuente el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.
- ✓ Entregas pequeñas. Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.
- ✓ Metáfora. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema, ayudando a la nomenclatura de clases y métodos del sistema).
- ✓ Diseño simple. Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.
- ✓ Pruebas. La producción de código está dirigida por las pruebas unitarias. Estas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema.
- ✓ Refactorización (*Refactoring*). Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para

facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo (Poppendieck, 2003).

- ✓ Programación en parejas. Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores).
- ✓ Propiedad colectiva del código. Cualquier programador puede cambiar parte del código en cuando lo desee.
- ✓ Integración continua. Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.
- ✓ 40 horas por semana. Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo (Gittins, y otros, 2001).
- ✓ Cliente in-situ. El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Éste es uno de los principales factores de éxito del proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.
- ✓ Estándares de programación. XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible.

1.5.5 Aplicaciones Web

En la ingeniería de software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. Las aplicaciones web son populares debido a lo práctico del

navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los *webmails, wikis, weblogs*, tiendas en línea y la propia Wikipedia que son ejemplos bastante conocidos de aplicaciones web (Luján , 2001).

1.6 Patrón de diseño: Modelo Vista Controlador

El modelo-vista-controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica del negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario (Anexo 3). Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento (Rivera, 2015)

El Modelo Vista Controlador es aplicable al desarrollo de cualquier aplicación independientemente del lenguaje de programación elegido. MVC consiste en dividir el código de una aplicación en capas (Fernández, 2012): **Modelo**: es todo acceso a datos y las funciones que llevan lo que llaman "lógica de negocio", o sea datos y reglas de negocio. Lleva un registro de las vistas y controladores del sistema. Cada acceso a datos se pone en su función individual porque, de esta forma, si se cambia de gestor de bases de datos este cambio sólo afecta a estas funciones, no al resto de la aplicación. Tener el modelo bien delimitado permite la existencia de varias aplicaciones que compartan el mismo. **Vista:** en una aplicación Web, es el HTML y lo necesario para convertir datos en HTML. O sea, muestra la información del modelo al usuario. Tienen un registro de su controlador asociado (normalmente porque además lo instancia). Pueden dar el servicio de "Actualización", para que sea invocado por el controlador o por el modelo. Tener la vista separada del controlador permite cambiar la

aplicación para que genere, en lugar de HTML, algo distinto (por ejemplo, WML), sin tener que tocar más que una parte completamente delimitada del código.

Controlador: es lo que une la vista y el modelo. Por ejemplo, son las funciones que toman los valores de un formulario, consultan la base de datos (a través del modelo) y producen valores, que la vista tomará y convertirá en HTML. En resumen, gestiona las entradas del usuario. Contiene reglas de gestión de eventos, del tipo “SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. De este modo, el código que “hace algo” está perfectamente separado del código dedicado a crear HTML.

1.7 Lenguajes de programación

PHP es un lenguaje de programación interpretado, originalmente diseñado para la creación de páginas web dinámicas. Sigue un estilo clásico ya que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones, etc. Está más cercano a JavaScript6 o al lenguaje C7, pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, lo que permite acceder a los recursos que este tenga, como es una base de datos y el resultado es enviado al navegador, el cual podría ser una página HTML o de cualquier otro tipo. (Autores, 2006)

PHP no necesita que el navegador lo soporte, es independiente de este, pero sin embargo para que sus páginas funcionen, el servidor donde están alojadas debe soportar este lenguaje, que tiene numerables ventajas sobre otros lenguajes de programación que se ejecutan de igual manera (como podrían ser los script CGI Perl), permitiendo intercalar las sentencias PHP en las páginas HTML. Es el lenguaje ideal para el desarrollo de este proyecto.

HTML: Hypertext Markup Language. Es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones

multimedia (gráficos, sonido). La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.), así como los diferentes efectos que se quieren dar (cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado.

No tiene nada que ver con un lenguaje de programación, quizás se parezca más al uso de un procesador de texto por la utilización de códigos de comienzo y finalizado de estilo. Además de resultar más sencillo, no se necesita ninguna herramienta de programación, compilador o similar, sino que cualquier editor de texto puede servir para realizar las páginas más sorprendentes. (Autores, 2015)

Java Scripts: es un lenguaje basado en objetos, muy parecido al Java, pero mucho más sencillo dada su simplicidad sintáctica y su manejabilidad, por lo que casi siempre está dentro de una página HTML. Habitualmente, se utiliza para activar procesos o desarrollar algoritmos que hagan tareas sencillas, (como calculadoras, almanaques, control de errores para formularios...) haciendo la web más viva y dinámica.

Como el HTML no permite activar procesos o desarrollar algoritmos, el Java Script viene muy bien para estas necesidades. No hay que olvidar que está muy limitado frente a lenguajes de programación tradicionales, sobre todo porque no permite la manipulación de ficheros (para evitar problemas de seguridad). Esto ahorra al servidor mucho trabajo pues el código se interpreta en el cliente (desde su navegador). Combinado con el acceso a bases de datos da mucho juego, pero sobre todo se usa para darle vida a las webs haciendo que cambien imágenes, que se procesen datos, entre otros.

CSS (*Cascading Style Sheets*): las hojas de estilo en cascada son un mecanismo que permite aplicar formato a los documentos escritos en HTML (y en otros lenguajes estructurados, como XML) separando el contenido de las páginas de su apariencia. Para el diseñador, esto significa que la información estará contenida en la página HTML, pero este archivo no

debe definir cómo será visualizada esa información. Las indicaciones acerca de la composición visual del documento estarán especificadas en el archivo de la CSS. Lo que posibilita crear páginas web de una manera más exacta. Gracias a esto el desarrollador es mucho más dueño de los resultados finales de la página.

Framework: en el desarrollo de software 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 otras aplicaciones para ayudar a desarrollar y unir los diferentes componentes de un proyecto, es decir, son soluciones completas que llevan incorporado herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución).

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. (Pressman, 2010)

Symfony es un completo *framework* diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de

gestores de bases de datos, como MySQL, PostgreSql, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. (Autores, 2009)

Symfony se utilizó porque se ajusta a los siguientes requisitos:

- Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de Propel, permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.
- Utiliza programación orientada a objetos, de ahí que sea imprescindible PHP 5.
- Sencillo de usar en la mayoría de casos, está más indicado para grandes aplicaciones Web que para pequeños proyectos.
- Aunque utiliza MVC (Modelo vista controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.

Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional:

- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo. Código fácil de leer que incluye comentarios de *phpDocumentor* y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.
- Una potente línea de comandos que facilitan generación de código ahorrándonos tiempo de trabajo.

JQuery es un framework de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, permitiendo manejar eventos,

desarrollar animaciones y agregar interacción con la tecnología AJAX, además de ser software libre y de código abierto. JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Otra gran ventaja de JQuery es que permite cambiar el contenido de la página web sin necesidad de recargarla, utilizando DOM y AJAX de manera extremadamente sencilla gracias a su sintaxis.

Sonata para administración

Bootstrap es un conjunto de herramientas gratuitas para crear sitios web y aplicaciones web. Contiene HTML y basados en CSS plantillas de diseño para tipografía, formas, botones, navegación y otros componentes de la interfaz, así como las extensiones de JavaScript opcionales.

En agosto de 2011, fue lanzado Twitter Bootstrap como código abierto. En febrero de 2012, fue el proyecto de desarrollo de GitHub más popular. Las características que dan motivo para utilizarlo en - Bootstrap es compatible con las últimas versiones de los principales navegadores. Su funcionamiento no es adecuado utilizarlo en navegadores antiguos como Internet Explorer 8.

Algunas de las características que dan motivo a su utilización:

- Desde la versión 2.0 también es compatible con el diseño de respuesta. Esto significa que el diseño de páginas web se ajusta dinámicamente, teniendo en cuenta las características del dispositivo utilizado (de escritorio, tableta, teléfono móvil).
- Bootstrap es de código abierto y está disponible en GitHub. Se anima a los desarrolladores a participar en el proyecto y hacer sus propias contribuciones a la plataforma.
- Recientemente, miembros de la comunidad han traducido la documentación del Bootstrap a varios idiomas, incluido el chino, español y ruso.

1.8 Sistemas de gestión de bases de datos

Los sistemas de gestión de bases de datos (SGBD) 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. La función general de un sistema gestor 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 una organización.

MySQL**:** es un gestor de base de datos sencillo de usar e increíblemente rápido sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas también en esos lenguajes. Es un sistema cliente-servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios y asegurando el acceso a solo los autorizados. Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad, utilizado por grandes corporaciones como Yahoo! Finance, Google, Motorola, entre otras. Es gratis para aplicaciones no comerciales. Aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multi-hilo. Dispone de interfaces de programación de aplicaciones en gran cantidad de lenguajes (C, C++, Java, PHP). Soporta hasta 32 índices por tabla. Implementa una gestión de usuarios y contraseñas que proporciona un buen nivel de seguridad en los datos. Es Software libre (licencia GNU GPL) y Open Source. Trabaja sobre muchas plataformas, incluida Windows, Mac Os X Server, Solaris, Linux, y muchas plataformas de UNIX. Acepta bloqueos y roles de usuario.

Se eligió MySQL, porque es uno de los servidores de bases de datos de código abierto más populares y conocidos del mundo, un sistema de

manejo de bases de datos con un gran nivel de estabilidad y facilidad de desarrollo que se integra fácilmente con el lenguaje de programación PHP. Dispone, además, de una arquitectura que lo hace extremadamente rápido y fácil de personalizar. Sumándole a todos estos beneficios, que es un servidor que se adecúa perfectamente a las exigencias del cliente.

1.8.1 Servidor web

Servidor Web: Apache HTTP Server el servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual. Apache tiene amplia aceptación en la red: en el 2005, Apache fue el servidor HTTP más usado, siendo el servidor empleado en el 48% de los sitios Web en el mundo. Sin embargo, ha sufrido un descenso en su cuota de mercado en los últimos años. (Esta dísticas históricas y de uso diario proporcionadas por Netcraft). La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache. Por lo tanto, las características a destacar para la utilización de este servidor son:

- Es modular.
- El servidor consta de una sección core y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor Web.
- Open-source. – Es un software distribuido y desarrollado libremente.
- Es gratuito.
- Multi-plataforma.

Presenta mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

1.9 Conclusiones parciales del capítulo.

En este capítulo se consultó la bibliografía y los antecedentes a la propuesta de investigación. Se definió la necesidad del desarrollo de una aplicación web que utiliza el lenguaje de programación PHP y el *framework* Symfony, el gestor de bases de datos MySQL y la metodología de desarrollo de software XP.

Capitulo II Análisis, diseño y construcción de la solución propuesta

2.1 Introducción.

En el capítulo se exponen los elementos necesarios para la descripción de la solución propuesta. A través de las Historias de Usuarios (HU) que acumulan la necesidad existente definida por el cliente, es llevado a cabo el análisis de los requerimientos. Para determinar si es factible el desarrollo del software se realizará un estudio de factibilidad.

2.2 Etapa de planificación.

Para lograr una eficiente planificación se inicia con una versión del prototipo inicial del problema y proporcionar así un buen comienzo a una solución eficaz. Con este objetivo y según las ideas del cliente sobre el software se desarrollarán las Historias de Usuarios (HU), mediante la cual se obtendrá un punto de partida para el resto de la planificación del proyecto. Igualmente se realizará un estimado de cada una de las entregas del proyecto y del tiempo, basándose en que la planificación inicial se podría afectar debido a cambios que pudiesen sufrir estos aspectos durante el desarrollo del proyecto.

2.2.1 Equipo de trabajo y roles.

Según la metodología XP, se hace necesario ver como se aplican las prácticas para entender más la composición del equipo del trabajo. Estas traducen valores mencionados en el capítulo anterior en actividades que un programador debe realizar diariamente (Rodríguez, 2006).

Algunos patrones a seguir por el equipo de trabajo:

- Se reafirma el principio de 40 horas de trabajo semanales, con algunos minutos de descanso cada dos o tres horas de trabajo continuo. Esto contribuye a evitar el cansancio mental de los desarrolladores, y con ello disminuye la probabilidad de errores.

- La retroalimentación con el cliente es un factor muy importante. Los reportes de fallos en el intercambio con este, pueden contribuir a que se detecten errores que pueden pasar por alto a vistas de los desarrolladores.
- El diseño debe ser simple y debe existir un estándar para la notación en el código, esto evita complicaciones en caso de que otra persona desee consultar el código fuente o tal vez los mismos desarrolladores después de algún tiempo.
- Se deben ir programando pequeñas versiones que van aumentando la dimensión poco a poco y en caso de existir fallos se lleva a cabo la reprogramación del código sin variar su funcionalidad.

Tabla 1 Equipo de trabajo y roles (Elaboración Propia)

Miembros	Roles
Jorge Martínez Pérez	Programador
Jorge Martínez Pérez, Jefa del Departamento de Ciencia Tecnología e Investigación	Encargado de Pruebas
Jefa del Departamento de Ciencia Tecnología e Investigación	Cliente

2.2.2 Historias de Usuario iniciales.

Según (Fuentes, 2015) para el establecimiento de las historias se utilizan dos escalas nominales que exponen tres categorías alta, media y baja las cuales significan el riesgo y la prioridad en la escala de riesgo y prioridad respectivamente.

A continuación, se muestran las escalas equivalentes a la prioridad en el negocio:

Alta: Asignada a las Historias de Usuario que corresponden a funcionalidades esenciales en el desarrollo del proyecto, a las que el cliente define como primordiales.

Media: Dada a las Historias de Usuario que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación directa sobre el proyecto que se esté desarrollando.

Baja: Se le otorga a las Historias de Usuario que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el proyecto en desarrollo.

Escala Nominal de Riesgo en Desarrollo:

Alta: Cuando para la implementación de la Historia de Usuario se considera la posible existencia de errores que lleven a inoperatividad del código.

Media: Cuando pueden aparecer errores en la implementación de la Historia de Usuario que puedan retrasar la entrega de la versión.

Baja: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Las HU son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

Son utilizadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si la aplicación cumple con lo que especifica la historia de usuario.

Estas ayudan en la comunicación entre el cliente y los desarrolladores y pueden ir cambiando a medida que avanza el proyecto y que el cliente vea nuevas posibilidades y soluciones. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia.

Con la aplicación se relacionan los usuarios: administrador del sistema, los profesores y la jefa del departamento de ciencia tecnología e investigación

En la tabla 2 se muestra una relación de las HU que se plantearon inicialmente.

Tabla 2 Historias de Usuario iniciales

No	Nombre	Prioridad	Riesgo	Iteraciones	Entrega
1	Diseño de la interfaz de usuario	Alta	Medio	1	1
2	Diseño y creación de la base de datos	Alta	Medio	1	1
3	Gestionar usuarios y roles	Alta	Medio	1	1
4	Gestionar facultad	Alta	Alto	1	1
5	Gestionar carrera	Alta	Alto	2	2
6	Gestionar categoría docente	Alta	Alto	2	2
7	Gestionar grado científico	Alta	Alto	2	2
8	Gestionar grupos	Alta	Alto	2	2
9	Gestionar bases de datos	Alta	Alto	2	2
10	Gestionar roles de publicación	Alta	Alto	3	3
11	Gestionar proyectos	Alta	Alto	3	3
12	Gestionar alcances de premios	Alta	Medio	3	3
13	Gestionar tipos de premio	Alta	Alto	3	3
14	Archivar publicación	Alta	Alto	3	3
15	Invalidar publicación	Alta	Alto	3	3
16	Generar reportes	Alta	Medio	3	3

A continuación, se muestran las HU en detalle para que se pueda comprender el proceso:

Tabla 3 HU 1

HISTORIA DE USUARIO	
Número: 1	**Usuario:** todos
Nombre de Historia: Diseño de la interfaz de usuario	
Prioridad en el negocio: Alta	**Riesgo en desarrollo**: Alta

Puntos estimados: 3	**Iteración asignada**: 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se muestra la vista de la interfaz principal en dependencia del rol del usuario que se ha registrado en el sistema.	

Tabla 4 HU 2

HISTORIA DE USUARIO	
Número: 2	**Usuario:**
Nombre de Historia: Diseño y creación de la base de datos	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 2	**Iteración asignada:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se diseña e implementa la base de datos en el gestor MySQL.	

Tabla 5 HU 3

HISTORIA DE USUARIO	
Número: 3	**Usuario**: Administrador
Nombre de Historia: Gestionar Usuarios y Roles	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Medio

Puntos estimados: 3	**Iteración asignada:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se permite al administrador gestionar los usuarios y asignar los roles correspondientes	

Tabla 6 HU 4

HISTORIA DE USUARIO	
Número: 4	**Usuario:** Administrador
Nombre de Historia: Gestionar facultad	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 1	**Iteración asignada:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se puede insertar, modificar y eliminar las facultades que están comprendidas por diferentes carreras.	

Tabla 7 HU 5

HISTORIA DE USUARIO	
Número: 5	**Usuario:** Administrador
Nombre de Historia: Gestionar carrera	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Alto

Puntos estimados: 1	**Iteración asignada:** 2
Programador responsable: Jorge Martínez Pérez	
Descripción: Se puede insertar, modificar y eliminar las carreras que están asociadas a las determinadas facultades.	

Tabla 8 HU 6

HISTORIA DE USUARIO	
Número: 6	**Usuario:** Jefa del departamento de CTI
Nombre de Historia: Gestionar categoría docente	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 1	**Iteración asignada:** 2
Programador responsable: Jorge Martínez Pérez	
Descripción: Se puede insertar, modificar y eliminar las categorías docentes.	

Tabla 9 HU 7

HISTORIA DE USUARIO	
Número: 7	**Usuario:** Jefa del departamento de CTI

Nombre de Historia: Gestionar grado científico	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 1	**Iteración asignada:** 2
Programador responsable: Jorge Martínez Pérez	
Descripción: Se puede insertar, modificar y eliminar los grados científicos.	

Tabla 10 HU 8

HISTORIA DE USUARIO	
Número: 8	**Usuario:** Jefa del departamento de CTI
Nombre de Historia: Gestionar grupo	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 2	**Iteración asignada:** 2
Programador responsable: Jorge Martínez Pérez	
Descripción: Permite insertar, modificar y eliminar grupos que están comprendidos por distintas bases de datos.	

Tabla 11 HU 9

HISTORIA DE USUARIO

Número: 9	**Usuario:** Jefa del departamento de CTI
Nombre de Historia: Gestionar base de datos	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 1	**Iteración asignada:** 2
Programador responsable: Jorge Martínez Pérez	
Descripción: Se puede insertar, modificar y eliminar las bases de datos las cuales están asociadas a grupos determinados.	

Tabla 12 HU 10

HISTORIA DE USUARIO	
Número: 10	**Usuario:** Jefa del departamento de CTI
Nombre de Historia: Gestionar roles de publicación	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 1	**Iteración asignada:** 3
Programador responsable: Jorge Martínez Pérez	
Descripción: Se puede insertar, modificar y eliminar los roles de publicación.	

Tabla 13 HU 11

HISTORIA DE USUARIO	
Número: 11	**Usuario:** Jefa del departamento de CTI
Nombre de Historia: Gestionar proyectos	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 1	**Iteración asignada:** 3
Programador responsable: Jorge Martínez Pérez	
Descripción: Se puede insertar y modificar los proyectos	

Tabla 14 HU 12

HISTORIA DE USUARIO	
Número: 12	**Usuario** Jefa del departamento de CTI
Nombre de Historia: Gestionar alcances de premios	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 1	**Iteración asignada:** 3
Programador responsable: Jorge Martínez Pérez	
Descripción: Se puede insertar, modificar y eliminar los diferentes alcances de premios	

Tabla 15 HU 12

HISTORIA DE USUARIO	
Número: 12	**Usuario** Jefa del departamento de CTI
Nombre de Historia: Gestionar tipos de premios	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 1	**Iteración asignada:** 3
Programador responsable: Jorge Martínez Pérez	
Descripción: Se puede insertar, modificar y eliminar los diferentes tipos de premios	

Tabla 16 HU 14

HISTORIA DE USUARIO	
Número: 14	**Usuario** Profesor
Nombre de Historia: Archivar publicación	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 3	**Iteración asignada:** 3
Programador responsable: Jorge Martínez Pérez	
Descripción: Se puede insertar un archivo cumpliendo con todos los requisitos necesarios y eliminarlo en caso de ser invalidado por la Jefa del departamento de CTI	

Tabla 17 HU 15

HISTORIA DE USUARIO

Número: 15	**Usuario** Jefa del departamento de CTI
Nombre de Historia: Invalidar archivo	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 1	**Iteración asignada:** 3
Programador responsable: Jorge Martínez Pérez	
Descripción: Se puede invalidar un archivo que contiene algún error en los parámetros requeridos.	

Tabla 18 HU 16

HISTORIA DE USUARIO	
Número: 16	**Usuario:** Jefa del departamento de CTI
Nombre de Historia: Generar reportes	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados: 1	**Iteración asignada:** 3
Programador responsable: Jorge Martínez Pérez	
Descripción: Se permite generar tablas con la información contenida en los archivos que cumplen los condiciones de filtrado dinámico.	

2.2.3 Planificación de iteraciones.

En la ilustración 1 se muestran las tres iteraciones y las HU que le corresponden.

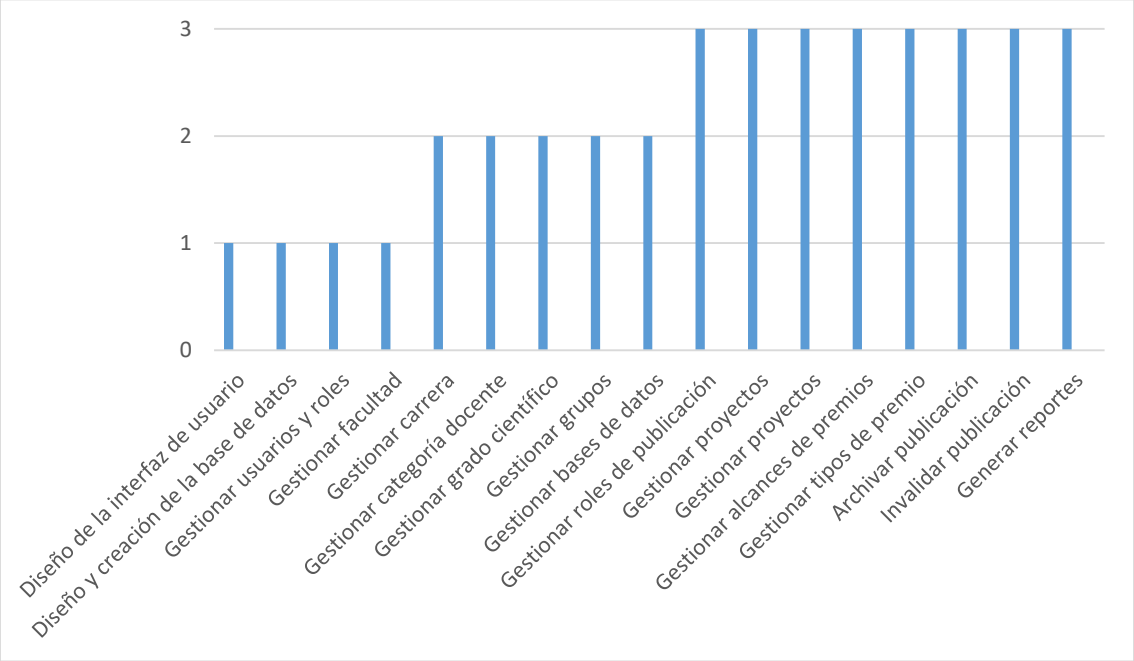

Ilustración 1 Planificación de las Iteraciones respecto a las HU (Elaboración propia)

2.2. 4 Plan de entregas.

La planificación de la entrega de acuerdo a la metodología XP propone agrupar una o varias iteraciones para que el cliente tenga versiones funcionales del sistema. En la ilustración 2 se muestra el Plan de Entregas.

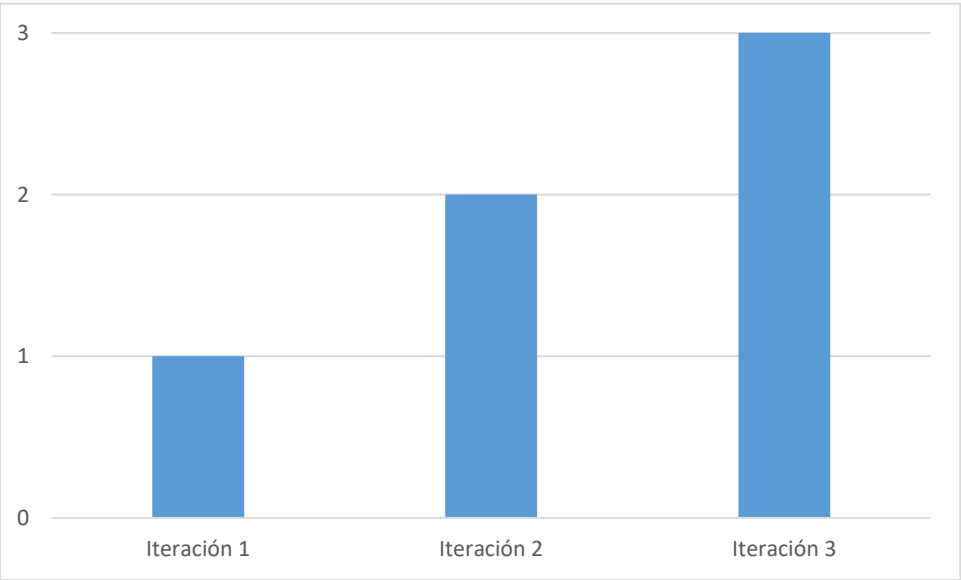

Ilustración 2 Plan de Entregas (Elaboración propia)

2.3 Etapas de diseño.

En XP solo se diseñan aquellas historias de usuario que el cliente ha seleccionado para la iteración actual por dos motivos: por un lado, se considera que no es posible tener un diseño completo del sistema y sin errores desde el principio. El segundo motivo es que, dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales del proyecto para luego modificarlo, se considera un desperdicio de tiempo.

Es importante resaltar que esta tarea es permanente durante la vida del proyecto partiendo de un diseño inicial que va siendo corregido y mejorado en el transcurso del proyecto.

2.3.1 Prototipo de interfaz de usuario.

A continuación, se muestran una ilustración de la interfaz de usuario, en el acápite de pruebas se pueden ver otras:

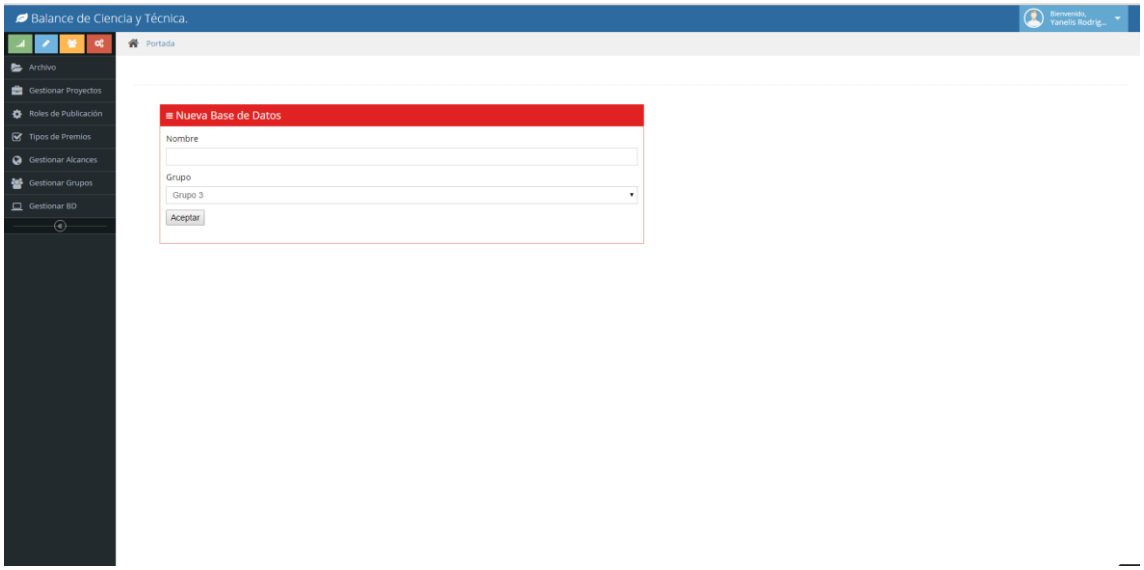

Ilustración 3. Para insertar una base de datos en el rol jefa de departamento de ciencia tecnología e investigación.

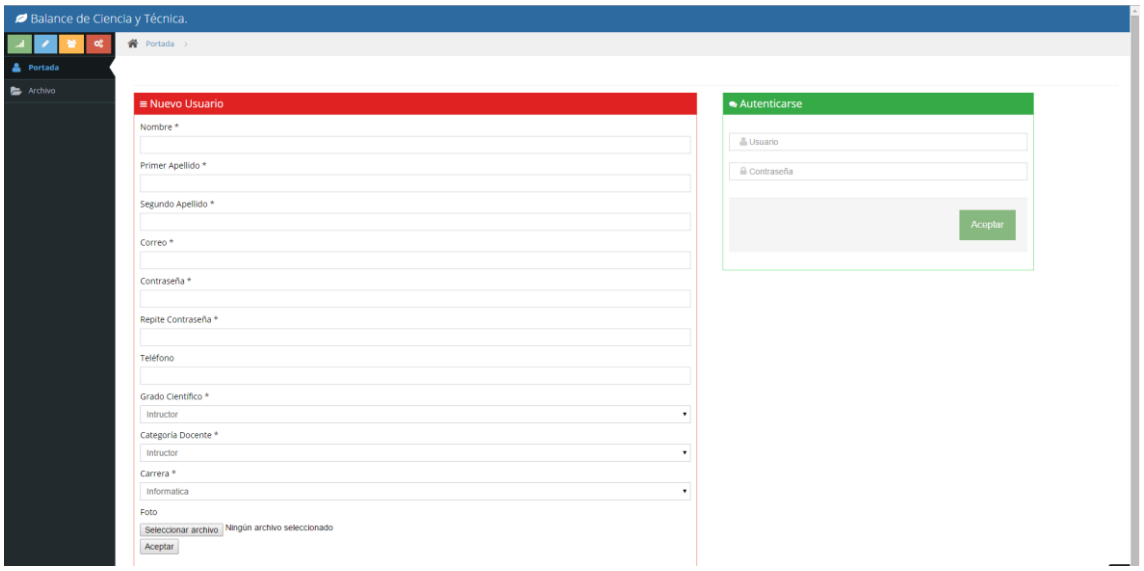

Ilustración 4 Interfaz de inicio de sección

2.3.2 Tareas a desarrollar.

Se muestra en la tabla la cantidad de tarea que corresponden a cada HU.

Tabla 19Tareas de Iteración (Elaboración propia)

No	Nombre HU	No	Tarea de Iteración	Itrc
1	Diseño de la interfaz de usuario.	1	Interfaz Principal	1
		2	Interfaz del Archivo	
		3	Otras Interfaces	
2	Diseño y creación de la base de datos.	4	Diseño de la base de datos.	1
		5	Creación de la base de datos	
3	Gestionar usuarios y roles.	6	Gestionar Usuario	1
		7	Autentificarse	
		8	Gestionar Roles	
4	Gestionar facultad	9	Insertar facultad	1
		10	Modificar facultad	
		11	Eliminar facultad	
5	Gestionar carrera	12	Insertar carrera	2
		13	Modificar carrera	
		14	Eliminar carrera	

6	Gestionar categoría docente	15	Insertar categoría docente	2
		16	Modificar categoría docente	
		17	Eliminar categoría docente	
7	Gestionar grado científico	18	Insertar grado científico	2
		19	Modificar grado científico	
		20	Eliminar grado científico	
8	Gestionar grupos	21	Insertar grupos	2
		22	Modificar grupos	
		23	Eliminar grupos	
9	Gestionar bases de datos	24	Insertar bases de datos	2
		25	Modificar bases de datos	
		26	Eliminar bases de datos	
10	Gestionar roles de publicación	27	Insertar roles de publicación	2
		28	Modificar roles de publicación	
		29	Eliminar roles de publicación	
11	Gestionar proyectos	30	Insertar proyectos	3
		31	Modificar proyectos	
12	Gestionar alcances de premios	32	Insertar alcance	3
		33	Modificar alcance	
		34	Eliminar alcance	
13	Gestionar tipos de premios	35	Insertar tipos de premios	3
		36	Modificar tipos de premios	
		37	Eliminar tipos de premios	
14	Archivar publicación	38	Insertar archivo	3
		39	Borrar archivo invalidado	
15	Invalidar publicación	40	Invalidar una publicación	3

16 Generar reporte	41	Generar reporte general	3
	42	Generar otros reportes	

2.3.3 Tareas de iteración

A continuación, se relacionan algunas tareas de iteración que tenían mayor peso en el desarrollo de este proyecto:

Tabla 20 TI 1

TAREA DE ITERACIÓN	
Número: 1	**No Historia:** 1
Nombre de Tarea: Interfaz principal	
Tipo de Tarea: Desarrollo	**Puntos estimados: 1**
Programador responsable: Jorge Martínez Pérez	
Descripción: Crear la interfaz visual principal de la aplicación	

Tabla 21 TI 5

TAREA DE ITERACIÓN	
Número: 5	**No Historia: 2**
Nombre de Tarea: Creación de la Base de Datos	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción:	

Se creó la Base de Datos empleando a partir de Visual Paradigma con el gestor MySQL

Tabla 19 TI 7

TAREA DE ITERACIÓN	
Número: 7	**No Historia: 3**
Nombre de Tarea: Autentificarse	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se otorgan los permisos de acuerdo al tipo de rol autentificado en el sistema.	

Tabla 20 TI 9

TAREA DE ITERACIÓN	
Número: 9	**No Historia: 4**
Nombre de Tarea: Insertar facultad	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se permite insertar una facultad.	

Tabla 21 TI 10

TAREA DE ITERACIÓN	
Número: 10	**No Historia: 4**
Nombre de Tarea: Modificar facultad	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se permite modificar una facultad.	

Tabla 22 TI 14

TAREA DE ITERACIÓN	
Número: 14	**No Historia: 5**
Nombre de Tarea: Eliminar carrera	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se permite eliminar una carrera.	

Tabla 23 TI 16

TAREA DE ITERACIÓN	
Número: 16	**No Historia: 6**
Nombre de Tarea: Modificar categoría docente	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Programador responsable: Jorge Martínez Pérez	

Descripción: Se permite modificar una categoría docente.

Tabla 24 TI 20

TAREA DE ITERACIÓN	
Número: 20	**No Historia: 7**
Nombre de Tarea: Eliminar grado científico	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se permite eliminar un grado científico.	

Tabla 25 TI 21

TAREA DE ITERACIÓN	
Número: 21	**No Historia: 8**
Nombre de Tarea: Insertar grupo	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se permite insertar un grupo.	

Tabla 26 TI 25

TAREA DE ITERACIÓN	
Número: 25	**No Historia: 9**
Nombre de Tarea: Modificar bases de datos	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1

Programador responsable: Jorge Martínez Pérez
Descripción: Se permite modificar una base de datos.

Tabla 27 TI 29

TAREA DE ITERACIÓN	
Número: 29	**No Historia: 10**
Nombre de Tarea: Eliminar roles de publicación	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se permite eliminar un rol de publicación.	

Tabla 28 TI 30

TAREA DE ITERACIÓN	
Número: 30	**No Historia: 11**
Nombre de Tarea: Insertar proyecto	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se permite insertar un proyecto.	

Tabla 29 TI 33

TAREA DE ITERACIÓN	
Número: 33	**No Historia: 12**

Nombre de Tarea: Modificar alcance	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se permite modificar un alcance	

Tabla 30 TI 36

TAREA DE ITERACIÓN	
Número: 36	**No Historia: 12**
Nombre de Tarea: Modificar tipos de premio	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se permite modificar un tipo de premio	

Tabla 31 TI 38

TAREA DE ITERACIÓN	
Número: 38	**No Historia: 12**
Nombre de Tarea: Archivar publicación	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Se permite archivar una publicación	

Tabla 32 TI 41

TAREA DE ITERACIÓN	
Número: 41	**No Historia: 13**
Nombre de Tarea: Reporte general	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Programador responsable: Jorge Martínez Pérez	
Descripción: Genera un reporta dados los diferentes parámetros del sistema que se han seleccionado.	

2.3.4 Diseño de la base de datos.

En la ilustración 4 se muestra el diagrama entidad relación que se diseñó y utilizó finalmente en el proyecto.

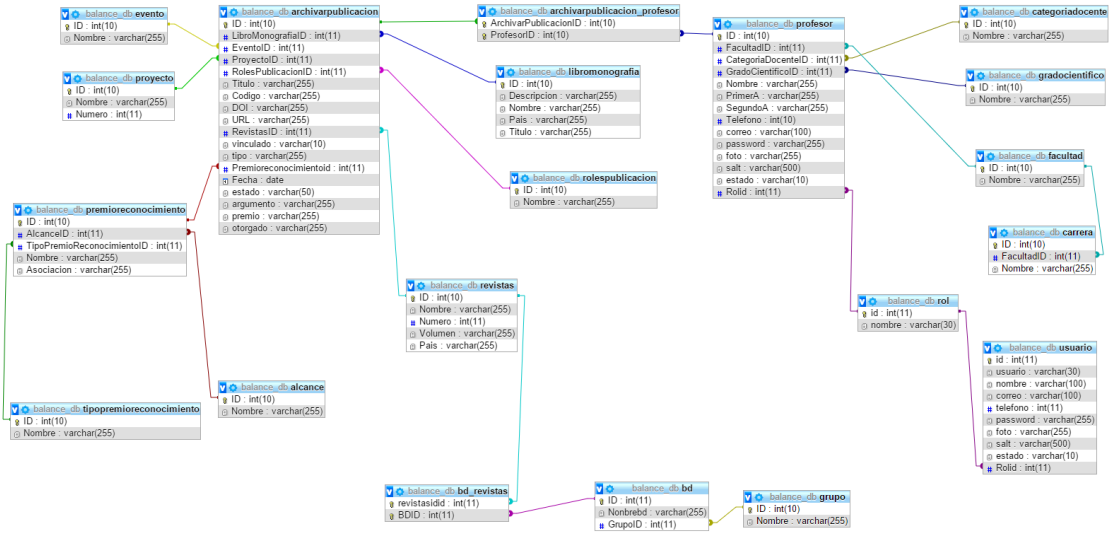

Ilustración 5 Modelo de entidad relación (elaboración propia)

2.3.5 Tarjetas de Clase, Responsabilidad y Colaboración.

A continuación, se muestras algunas tarjetas de Clase, Responsabilidad y Colaboración (CRC) creadas:

Tabla 33 Tarjeta CRC Usuarios

Profesor	
Superclase:	
Subclase:	
Descripción: En esta clase se guardan los elementos referidos a los usuarios profesor.	
ATRIBUTOS	
idUsuario	int
nombreprofesor	string
primerapellido	string
segundoapellido	string
facultad	string
gradocientífico	string
categoriadocente	string
correo electronico	string
telefono	int
Contraseña	string
foto	string

Tabla 34 Tarjeta CRC Pedidos

Libro o monografía
Superclase:
Subclase:
Descripción:

En esta clase se guardan los elementos referidos a los libros o monografías	
ATRIBUTOS	
Idlibromonografia	int
descripción	string
editorial	int
país	string

2.5 Conclusiones parciales del capítulo

En este capítulo se plantean las etapas necesarias para desarrollar el software según la metodología XP, con la excepción de las pruebas funcionales. Se escriben las historias de usuario y tareas iniciales que se agrupan en iteraciones y entregas. Se pudo concretar al final de las tres iteraciones la entrega del sistema informático completamente en funcionamiento.

Capítulo III Validación de la Solución Propuesta

En este capítulo se comprueba el cumplimiento de los requerimientos iniciales, con este propósito se desarrollan los casos de prueba, las estrategias de prueba y de forma similar se estudian las pruebas de caja negra a través de las cuales se efectúan las pruebas de aceptación y se analizan los resultados obtenidos. Un proceso de desarrollo de software es la descripción de una secuencia de actividades que deben ser seguidas por un equipo de trabajadores para generar un conjunto coherente de productos, uno de los cuales en el programa del sistema deseado. El objetivo básico del proceso es hacer predecible el trabajo que se requiere: predecir el costo, mantener un nivel de calidad y pronosticar el tiempo de desarrollo (Drake, 2008)**.** Muchas veces no se dispone de herramientas, ni siquiera de metodologías, que permitan transformar el software

ordinario en otro que sea fiable y fácil de mantener. Los sistemas software medianamente grandes suelen estar "plagados" de errores, y realizar cambios en ellos es, cuando menos, una tarea arriesgada (Gutierrez, 2011). A causa de la falta de comunicación y el poco intercambio, el desarrollo del software lleva aparejado un conjunto de actividades con el fin de asegurar su calidad (Rivero, 2015). Se presenta un software que responde a los objetivos planteados en la introducción de este trabajo, y este capítulo estará dedicado esencialmente a la validación dicho software.

3.1 Pruebas

3.1.1 Pruebas de aceptación

Las pruebas de aceptación son las realizadas por el cliente y usuarios finales de la aplicación. Permiten probar las funcionalidades que exige el cliente. Luego de haber superado las pruebas de aceptación podrá considerarse que la aplicación es apta para el uso y despliegue dentro del proyecto.

Las pruebas persiguen como objetivo, llevar a cabo el proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar al menos un error no descubierto hasta entonces (Pressman, 2010).

Como resultado de las pruebas de aceptación se obtendrán artefactos descritos en tablas, estas contarán con los siguientes campos:

1. Código: Constituye un indicador de la prueba realizada y es sugerente al nombre de la prueba a la que hace referencia.

2. HU: Contiene el nombre de la historia de usuario a la que hace referencia la prueba a realizar.

3. Nombre: Nombre que se le asigna a la prueba a realizar.

4. Descripción: Se detalla la funcionalidad que se desea probar.

5. Condiciones de Ejecución: Muestra las condiciones necesarias y, que deben ser satisfechas antes de efectuar la ejecución del caso prueba para desarrollar el mismo y obtener los resultados esperados.

6. Entradas / Pasos de Ejecución: Se describen los pasos seguidos en el desarrollo de la prueba, para ello se consideran todas las entradas que hace el usuario, con el propósito de conocer si se obtiene el resultado esperado.

7. Resultado esperado: Se define el resultado que se espera obtener con la prueba realizada.

8. Evaluación de la prueba: Una vez efectuada la prueba y obtenido su resultado se expone una valoración de la misma. Se clasifican en:

Satisfactoria: Cuando el resultado de la prueba es exactamente el esperado por el usuario.

Parcialmente satisfactoria: cuando el resultado no es completamente el esperado por el cliente o usuario de la aplicación y muestra resultados erróneos o fuera de contexto.

No satisfactoria: Cuando el resultado de la prueba realizada genera un error de codificación en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, trayendo como consecuencia que la funcionalidad requerida por el cliente no tenga resultado, lo que invalida también la HU.

Las pruebas de aceptación se llevan a cabo al efectuar los pasos siguientes:

1. Se redactan los casos de prueba teniendo en cuenta el orden de las historias de usuario y los niveles de prioridad dados a las funcionalidades.

2. Se planifica con el cliente de cuándo y cuáles pruebas serán llevadas a cabo.

3. Se reúnen los miembros del proyecto seleccionados para realizar las pruebas.

4. Se completan cada uno de los campos de las tablas de las pruebas de aceptación con el resultado de la prueba.

Las pruebas de aceptación, alternadas con vistas de los mensajes de errores que fueron detectados a medida que se fueron realizando las mismas o que forman parte de la validación, se muestran seguidamente:

Tabla 35 PA 1

Pruebas de aceptación

Número Caso de Prueba: 1	**No Historia**:1
Nombre Caso de Prueba: Test Interfaz Principal del rol Jefa del Departamento de ciencia técnica e investigación.	
Descripción: Verificar que se muestren las interfaces visuales implementadas	
Condiciones de ejecución: Esté corriendo la aplicación.	
Entradas: Interfaces de la aplicación.	
Resultado esperado: Se muestren las interfaces visuales de la aplicación.	
Evaluación: Prueba satisfactoria	

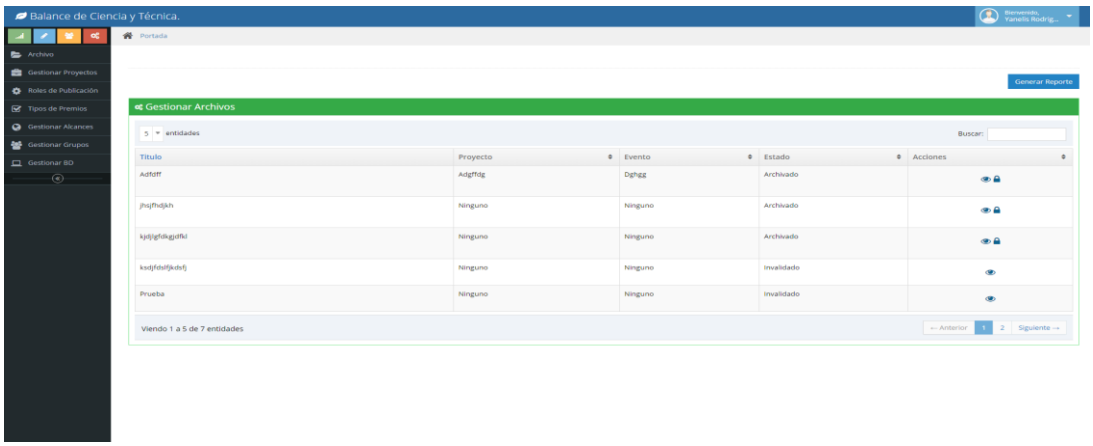

Ilustración 6 Resultado de caso de prueba No. 1

Tabla 36 PA 2

Pruebas de aceptación	
Número Caso de Prueba: 2	**No Historia:** 2

Nombre Caso de Prueba: Test Base de Datos
Descripción: Verifica el funcionamiento de la base de datos
Condiciones de ejecución: Estar conectado a la base de datos.
Entradas: Valores para leer o escribir en la base de datos por ejemplo: grupos, usuario, contraseña, etc.
Resultado esperado: Se muestran o guardan los datos correctamente.
Evaluación: Prueba satisfactoria

Tabla 37 PA 3

Pruebas de aceptación	
Número Caso de Prueba: 3	**No Historia**: 3
Nombre Caso de Prueba: Test Autentificar	
Descripción: Verificar que se autentifique un usuario correctamente en el sistema.	
Condiciones de ejecución: Estar conectado a la Base de Datos	
Entradas: Nombre de usuario y contraseña	
Resultado esperado: Se otorgan al usuario correspondiente los permisos que le corresponden a su rol.	
Evaluación: Prueba satisfactoria	

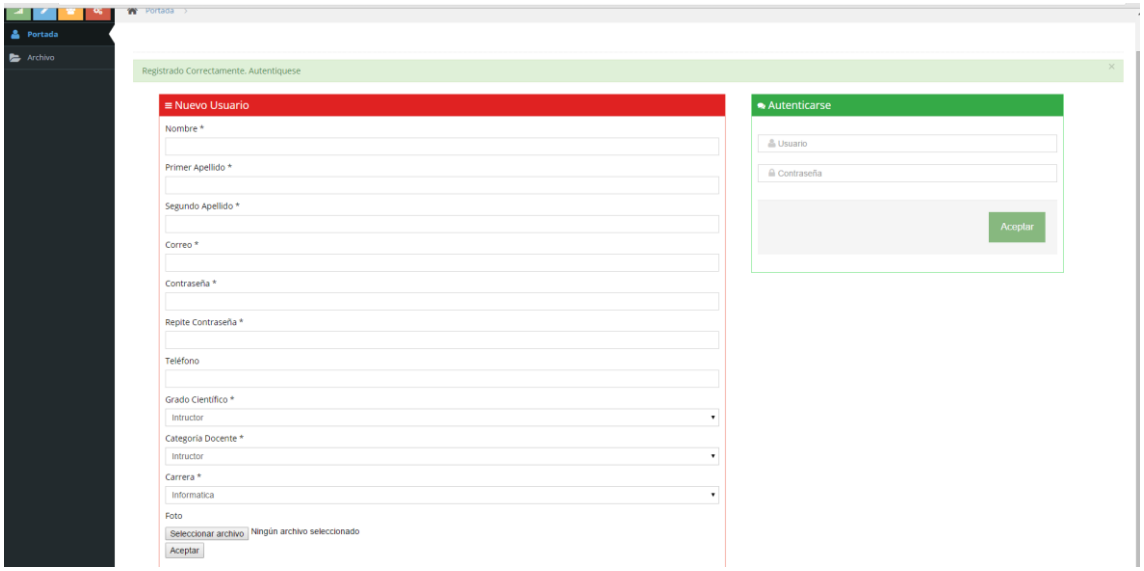

Ilustración 7 Formulario para registrarse.

Tabla 38 PA 4

Pruebas de aceptación	
Número Caso de Prueba: 4	**No Historia:**10
Nombre Caso de Prueba: Test gestionar rol de publicación	
Descripción: Verificar que se pueda editar un rol de publicación correctamente.	
Condiciones de ejecución: Estar conectado a la base de datos.	
Entradas: Elementos del rol de publicación.	
Resultado esperado: Se edita correctamente un rol de publicación	
Evaluación: Prueba satisfactoria	

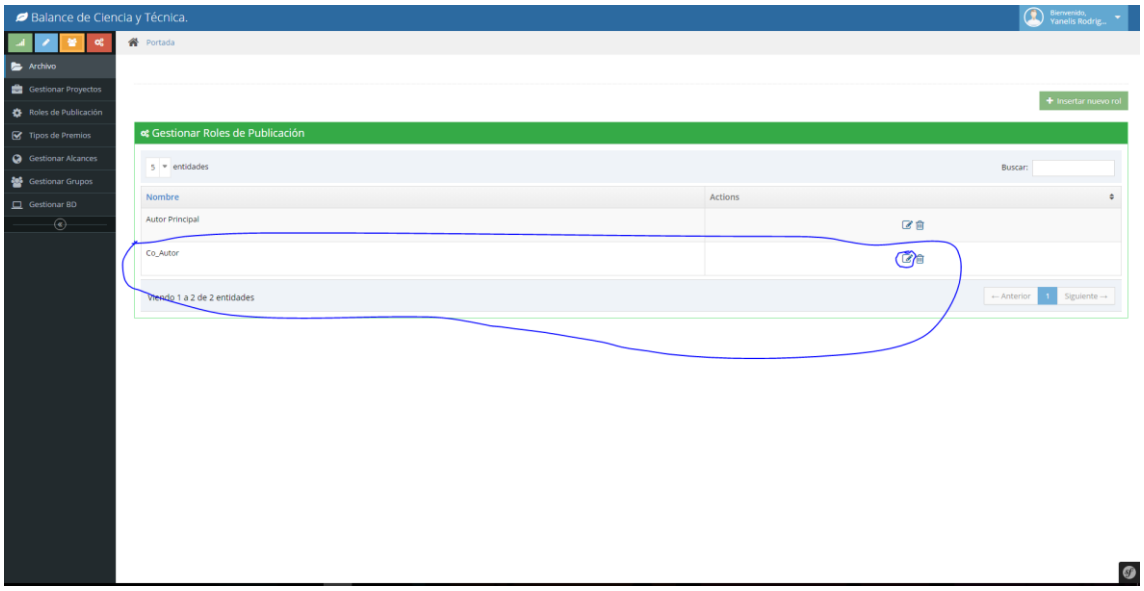

Ilustración 8 Resultado de caso de prueba No. 4 (elaboración propia).

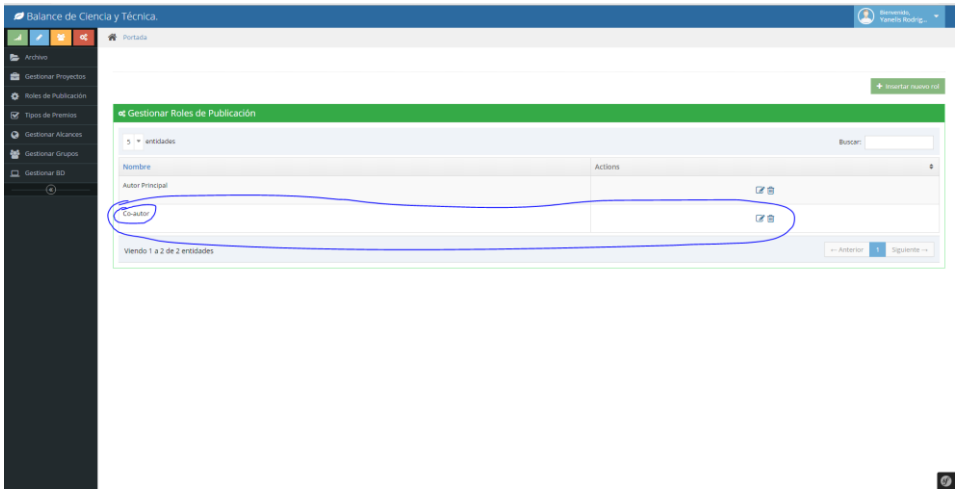

Ilustración 9 Resultado de caso de prueba No. 4 (elaboración propia).

3.2 Análisis de los resultados

Después de desarrollar todo un proceso de pruebas con un nivel medio de sencillez se lograron resultados satisfactorios, pues tras la detección de diferentes errores, obtenidos fundamentalmente con las realizadas, se solucionaron varios problemas que impedían el cumplimiento de los requisitos fundamentales del sistema en cuestión.

Las primeras pruebas fueron planeadas y ejecutadas en módulos individuales del programa y a medida que fueron avanzando se desplazaron a módulos

integrados, hasta que finalmente llegaron al sistema completo y se logró obtener un software cuyas funciones se encuentra en correspondencia con las especificaciones acordadas y que además cumple con los requerimientos de rendimiento.

El desarrollo del sistema cumple las expectativas trazadas al inicio del proyecto y satisface al cliente en su totalidad ya que se logra facilitar el proceso de desarrollo del balance de ciencia de la Universidad de Matanzas.

3.3 Beneficios tangibles e intangibles. Estimación del costo

Bohem plantea una fórmula que en es aplicable en general a los procesos ágiles, lo cual se adecua perfectamente a este proyecto de desarrollo de software que utilizó la metodología XP.

La fórmula especificada plantea que:

Costo Total (CT) = Tiempo de Desarrollo (TD) * Salario Mensual (SM) * Cantidad de Hombres (CH)

TD = 24 semanas, lo cual se obtuvo sumando los puntos estimados de las historias de usuarios, agregando el tiempo dedicado a las pruebas. Es aproximadamente 6 meses.

SM = $ 500, teniendo en cuenta el salario promedio de un ingeniero en Cuba.

CH = 1, el autor de este trabajo.

CT = $ 500 * 6 * 1

CT = $ 3000

Cuando se realiza un cálculo del costo de una aplicación, es preciso realizar un análisis del mismo en relación a los beneficios tangibles e intangibles. Se debe aclarar que la aplicación fue adquirida de manera gratuita. Evidentemente los beneficios reflejados a lo largo de este trabajo son superiores al costo de desarrollo de la aplicación.

3.4 Conclusiones parciales

En este capítulo se mostraron los elementos de prueba y los casos de prueba que se le aplicaron, se explicaron las estrategias de prueba. Las pruebas se convierten en una herramienta de desarrollo, son un paso imprescindible mediante la verificación del buen funcionamiento de un software.

Como resultados finales se obtuvo una aplicación web, con una apariencia agradable y fácil de usar. La planificación inicial se cumplió a un 100 %, no existieron gastos para la entidad, pues el software se realizó sin afectar el horario de trabajo del programador, se utilizaron las herramientas más actuales para su desarrollo, y el plan de entrega fue cumplido con éxito. El cliente quedó complacido con el trabajo.

La validación realizada de la propuesta de una aplicación web para facilitar la realización del balance de ciencia y técnica de la Universidad de Matanzas tuvo resultados positivos, lo que fue posible constatar mediante un análisis que permitió validar los niveles de aceptación de este resultado informático sobre la base de la relación cliente – usuario. En tal sentido pudo corroborarse, además, los beneficios que se alcanzan con la puesta en práctica de dicha aplicación web, lo que fue resultado principal de la presente investigación.

Conclusiones

Se realizó un estudio profundo del proceso de balance de ciencia y técnica en la Universidad de Matanzas Camilo Cienfuegos. Se analizaron las herramientas para el desarrollo de la aplicación web en apoyo al balance de ciencia y técnica, se seleccionó el framework Symfony y el gestor MySQL. Se utilizó de la metodología XP para el desarrollo de la aplicación web. Se validó la propuesta implementada con las pruebas de la metodología XP y el cumplimiento de los requerimientos iniciales. Se le dio solución al problema científico planteado con el desarrollo de esta aplicación web.

Bibliografia

- Beck, K. 2000. Extreme Programming Explained: Embrace Change. s.l. : Addison-Wesley. Pearson Education, 2000.

- Blanch, Bataller, Daniel. 2010. Implementación de un Sistema de Reservas para una Agencia de Viajes usando J2EE y prácticas de Desarrollo Ágil. España : Universidad Obrera de Catalunya.Ingeniería Técnica en Informática de Gestión, 2010.

- Cao, Lan, y otros. 2004. How Extreme does Extreme Programing Have to be? Adapting XP Practices to Large-scale Projects. s.l. : Hawaii International Conference on System Sciences, 2004. 0-7695-2056-1.

- Calero, Manuel Solís. 2003. Una explicación de la programación extrema(XP). [En línea] 2003. [Citado el: 3 de Mayo de 2017.] www.apolosoftware.com.

- Cubel, Navarro y Jose, María. 2012. Extreme Programming Laboratorio de Sistemas de Información. Valencia , España : Universidad Politécnica de Valencia, 2012.

- Fernández, D G. 2012. Sistema Informático en plataforma libre para el desarrollo de las técnicas de comercio electrónico en la Tecnología. s.l. : OTEC, 2012.

- Fuentes, J. (2015). Desarrollo de Softwre Ágil. Extremme Programming y Scrum.

- Giraldo, Z. 2006. Herramientas de Ingeniería del Software. 2006.

- Gittins, Robert y Hope, Sian. 2001. A study of Human Solutions in eXtreme Programming. Bangor Reino Unido : Shool of Informatics University of Wales Bangor, 2001.

- Goto, Takaaki, Tsuchida, Kensei y Nishino, Tetsuro. 2014. An Extreme Programming Method for Innovative Software Based on Systems Design and its Practical Study. Japón : International Journal of Software Engineering & Applications (IJSEA), 2014. Vol. 5, 5.

- Gutierrez, Demián. 2011. Métodos de desarrollo de software. Venezuela : Universidad de los Andes, 2011.

- Highsmith, J y Cockburn, A. 2001. Agile Software Development: The Business of Innovation. s.l. : IEEE Computer, 2001.

-Hurtado, Julio Ariel y Bastiarrica, Cecilia. 2005. Modelo de Procesos, Calidad y Mejoramiento. s.l. : Proyecto SIMEP-SW, 2005.

-Iván F. Palomo, Carlos G. Veloso, and Rodolfo F. Schmal. Información tecnológica sistema de gestión de la investigación en la universidad de talca, chile, 2007. URL

http://www.scielo.cl/scielo.php?pid=S071807642007000100014&script=sci_arttext.

- Jeffries,R, Anderson,A y Hendrickson, C. 2001. Extreme Programming Installed. s.l. : Addison Wesley, 2001.

- José H, Canós, Letelier, Patricio y Penadés, Carmen. 2003. Metodologías Ágiles en el Desarrollo de Software. España : Universidad Politécnica de Valencia, 2003.

-Joskowicz, José. 2008. Reglas y Prácticas en Extreme Preogramming. 2008.

-Letelier, Patricio, Penadés, María Carmen y Canós, José H. 2006. Metodologías Ágiles en el Desarrollo de Software. Valencia, España : Universidad Politécnica de Valencia, 2006.

-Luján , Mora Sergio. 2001. Clientes Web. s.l. : Editorial Club Universitario, 2001.

- Monguel, E.A. 2005. Metodología de la Investigación. México : Universidad Juárez Autonoma de Tabasco, 2005.

-Poppendieck, M. 2003. Lean Software Development: An Agile Toolkit for Software Development Managers. s.l. : Addison Wesley, 2003.

- Pressman, R. 2010. Ingeniería de Software. Un enfoque práctico. 2010.

-Reynoso, Billy Carlos. 2012. Métodos Ágiles en Desarrollo de Software,Introducción a la Arquitectura de Software. Universidad de Buenos Aires : s.n., 2012.

-Rivera, Yeniset. 2015. Sistema de control de Activos Fijos en la Entidad de Cultura. Matanzas : Universidad de Matanzas, 2015.

-Rivero, Omar Muñiz. 2015. Sistema informático para la automatización del análisis. Matanzas, Cuba : Trabajo para optar por el Título de Ingeniero en Informática.Universidad de Matanzas , 2015.

-Robles, G y Ferrer, Jorge. 2002. Programación Extrema y Software Libre. [En línea] 20 de marzo de 2002. [Citado el: 9 de enero de 2017.] http://es.tldp.org/Presentaciones/200211hispalinux/ferrer/robles-ferrer-ponencia-hispalinux-2002.pdf.

-Rodríguez, Rivero, Raymol Samuel. 2016. C# Tutorial. [En línea] 2016. http://www.csharpya.com.ar.

- SIICYT, March 2014. URL http://www.siicyt.gob.mx/siicyt/quienesSomos.do?pSel=%27%27.

-Wake, William C. 2000. Extreme Programming Explored. s.l. : Copyright 2000, 2000.