

Universidad de Matanzas



Facultad de Ciencias Técnicas

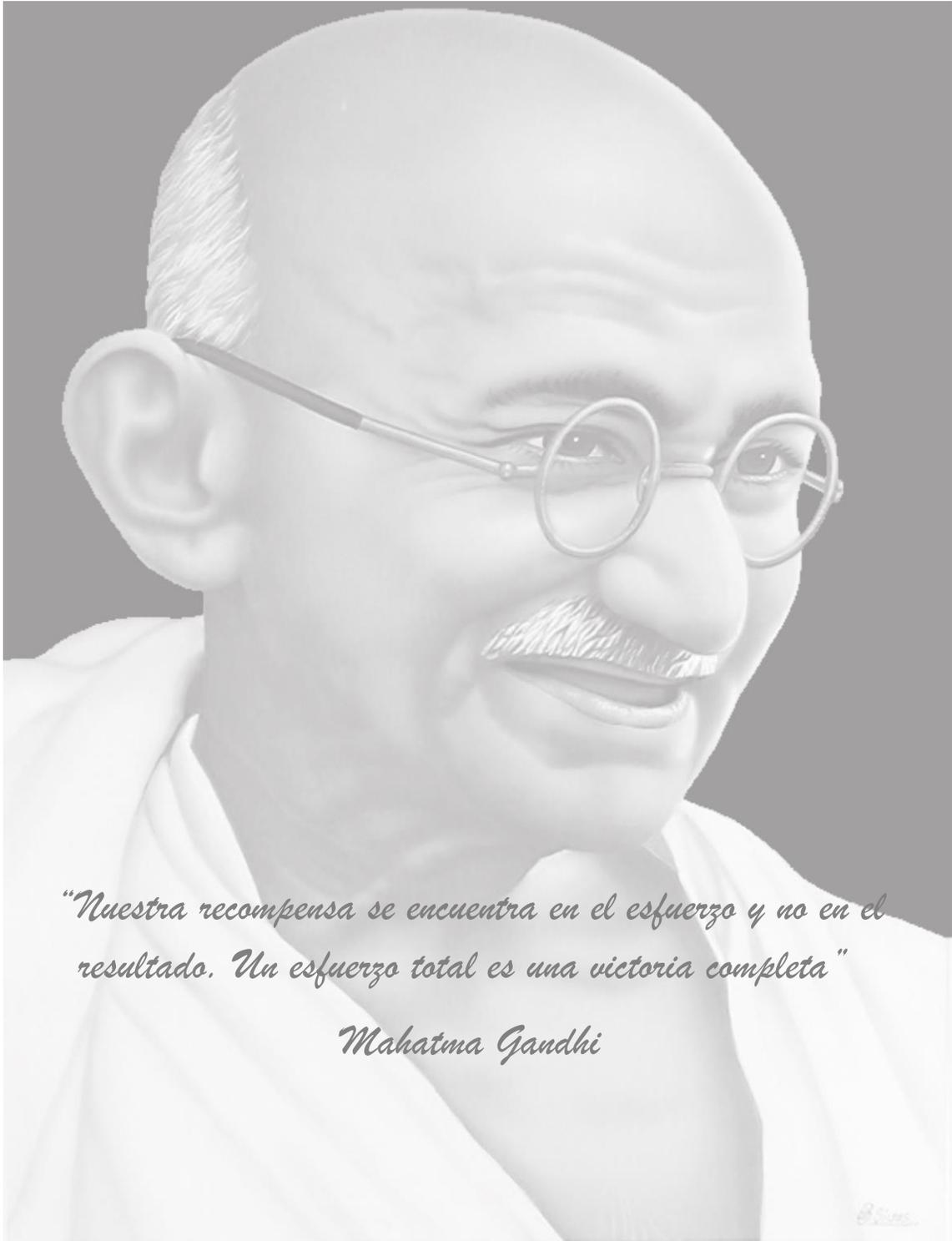


Ingeniería Informática



Trabajo de Diploma para optar el título de Ingeniero Informático
Título: Sistema web para la gestión de los Programas de Formación
Doctoral.

Autor: Lorna Ylenia Blanco González
Tutor(es): Dr.C. Walfredo González Hernández.
Matanzas, Cuba
Junio, 2019



"Nuestra recompensa se encuentra en el esfuerzo y no en el resultado. Un esfuerzo total es una victoria completa"

Mahatma Gandhi

Dedicatoria

A mi familia por estar siempre presente, su amor incondicional y por guiarme por el camino correcto.

Gracias por creer en mí en todo momento y darme fuerzas para seguir adelante, por ayudarme a luchar por mis sueños. Simplemente les doy las gracias por existir. Los quiero mucho.

Agradecimientos

A mi esposo por formar parte de mi vida, por darme las fuerzas y los abrazos cuando más los necesite, por acompañarme incondicionalmente en todo momento sin importar las largas horas espera. Por complacer cada uno de mis caprichos sin importar los obstáculos, por formar parte de esta trayectoria de muchos sacrificios. Infinitas gracias a ti por ser de oro para mí.

A mi abuela Gladys, a mi mamá y a mi hermana, por formar parte de mi día a día, por guiarme, exigirme, educarme, quererme y consentirme. Por estar presentes en todo momento apoyándome y consintiéndome.

A mi papa, por ser mi papa y a pesar de la lejanía siempre estar presente y orgulloso, sobre todas las cosas, de lo que me he convertido.

A mis abuelos paternos, por quererme tanto y estar siempre desde pequeña pendientes de mi día a día formando parte en todos los momentos de mi vida. Por consentirme, y ser hasta magos para mí.

A mi familia en general que de una forma u otra siempre me han apoyado en el transcurso de mi vida.

A mi amiga, y más que eso mis hermanas, Rocío. Gracias por compartir conmigo tantos momentos inolvidables, estar pendiente, apoyarme, preocuparte, aconsejarme y querer siempre lo mejor para mí. Gracias por no faltarme nunca y darme el privilegio de tu amistad.

A mis amigos habaneros, Laura, Jerny y Genry que a pesar del poco tiempo han formado parte de mi vida y con los cuales he pasado excelentes e inolvidables momentos.

A todos los compañeros de estudio que he tenido y han formado parte de este largo camino. En especial a los llamados matanceros, Laura y Keila por siempre avisarme ayudarme e informarme.

A todos los educadores que han contribuido con mi formación profesional.

A mi tutor, Walfredo, por tanta ayuda, dedicación, preocupación, sentido de pertenencia y darme las fuerzas para continuar durante el desarrollo de este trabajo.

A todos los que de una forma u otra han contribuido a la realización del presente trabajo.

A todos aquellos que no he mencionado pero que de una forma u otra han formado parte de este largo trayecto.

A todos, mil gracias.

Declaración de Autoría

Yo, Lorna Ylenia Blanco González, declaro que soy el único autor de este trabajo y autorizo a la Universidad de Matanzas Sede: " Camilo Cienfuegos ", especialmente a la Facultad de Ciencias Técnicas, a que hagan el uso que estimen pertinente de él.

Para que así conste firmo la presente a los ____ días del mes de junio del año 2019.

Firma del Autor

Lorna Ylenia Blanco González

Firma del Tutor

Walfredo González Hernández



Universidad de Matanzas
Vicerrectoría de Investigación y Post grado
Departamento de Grado Científico

Matanzas, 6 de junio de 2019

“Año 61 de la Revolución”

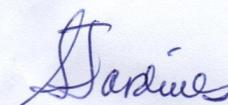
AVAL

El software “Sistema web para la gestión del proceso de formación doctoral” realizado por la estudiante Lorna Ylenia Blanco González, es de gran utilidad para el departamento de Grado Científico de la Universidad de Matanzas, toda vez, que con el desarrollo tecnológico, la informatización de los procesos se hace más necesaria. Hasta este momento no ha existido una plataforma interactiva que permita viabilizar y gestionar el proceso desde que ingresa el aspirante, hasta que concluye finalmente la defensa.

El diseño en el que está concebido el mismo es agradable, fácil de trabajar, sencillo y útil, ya que permite acceder a una gran cantidad de información y realizar adecuaciones según indiquen las normativas que rigen la formación doctoral en el MES.

Sugerimos jerarquizar los programas de formación doctoral con gestores para cada uno de ellos e incluir otros ítems en la base de datos de los aspirantes, lo que permitiría filtrar informaciones necesarias del propio proceso.

Atentamente,


Dr.C Sonia Jardines González

J'Dpto de Grados Científicos UM



Opinión del tutor del Trabajo de Diploma

TÍTULO DE LA TESIS: Sistema web para gestión de los Programas de Formación Doctoral

Autor: Lorna Ylenia Blanco González

Tutor(es): Dr.C. Walfredo González Hernández.

Tabla de contenido

Introducción	1
Capítulo 1 Fundamentación teórica.....	6
1.1 Introducción	6
1.2 Conceptos fundamentales asociados al dominio del problema.	6
1.3 Objeto de estudio.....	7
1.4 Proceso de formación doctoral	8
1.4.1 Diagrama de Procesos del Negocio.....	13
1.4.2 Antecedentes del trabajo.....	13
1.5 Metodología de desarrollo, herramientas y tecnologías.....	13
1.5.1 Metodología de desarrollo.....	14
1.5.2 Tecnologías y Herramientas.....	16
1.6 Conclusiones parciales del Capítulo:.....	23
Capítulo 2 Solución Teórica del Problema Científico	24
2.1. Introducción	24
2.2. Flujo de trabajo de los procesos a informatizar en la entidad correspondiente.	24
2.3. Requisitos del software.....	25
2.3.1. Requisitos funcionales.....	25
2.3.2. Requisitos no funcionales	26
2.4. Fase de planificación	29
2.4.1. Historias de Usuario.....	29
2.4.2. Plan de entregas.....	30
2.4.3. Iteraciones	31
2.4.4. Plan de iteraciones.....	31
2.5. Análisis de los costos.....	31
2.5.1. Beneficios tangibles e intangibles.....	36
2.5.2. Análisis de Costos.....	36
2.6. Fase de diseño.....	36
2.6.1. Arquitectura del sistema.....	36

2.6.2. Modelado del diseño.....	39
2.6.3. Diseño de tarjetas de Clase, Responsabilidad y Colaboración (CRC).....	40
2.7. Implementación.....	41
2.7.1. Tareas de ingeniería por historia de usuario.....	41
2.8. Conclusiones del capítulo	42
Capítulo 3. Validación de la solución propuesta.....	43
3.1. Introducción	43
3.2. Pruebas al software.....	43
3.3. Pruebas funcionales.....	43
3.4. Pruebas de usabilidad	55
3.5. Pruebas de seguridad.....	56
3.6. Pruebas de aceptación	58
3.7. Análisis de los resultados obtenidos.	60
3.8. Conclusiones del capítulo.....	60
Conclusiones Generales	62
Recomendaciones.....	63
Bibliografía Referenciada.....	64
Anexos	67
Anexo 1. Mapa de procesos del negocio.....	67
Anexo 2. Mapa de procesos del negocio implementado	69
Anexo 3. Diferencias entre las metodologías SCRUM y XP.	70
Anexo 4. Resumen de HU	71
Anexo 5. Plan de Iteraciones.....	73
Anexo 6. Diagrama Entidad-Relación.....	74
Anexo 7. Lista de Chequeo.....	74

Índice de Ilustraciones

Ilustración 1 Diagrama de despliegue.....	40
Ilustración 2 Configuración del software Vega.....	57
Ilustración 3 Resultado del software de testeo Vega.....	57
Ilustración 5 Mapa de Procesos. Fuente: Elaboración del Autor.....	67
Ilustración 6 Mapa de Procesos II. Fuente: Elaboración del Autor.	68
Ilustración 7 Mapa de procesos del negocio implementado.....	69
Ilustración 8 Plan de iteraciones. Fuente: Elaboración del Autor.	73
Ilustración 9 Diagrama Entidad-Relación. Fuente: Elaboración del Autor.	74

Índice de Tablas

Tabla 1 Equipo de Trabajo Fuente: Elaboración del Autor.	24
Tabla 2 Historia de Usuario Fuente: Elaboración del autor.	30
Tabla 3 Tarjeta CRC Fuente: Elaboración del Autor.	40
Tabla 4 Tarea #24 Insertar Programa. Fuente: Elaboración del Autor.	41
Tabla 5 Tarea #25 Modificar Programa. Fuente: Elaboración del Autor.	41
Tabla 6 Plan de Pruebas Fuente: Elaboración del Autor.....	44
Tabla 7 Clases de equivalencia. Fuente: Elaboración del Autor.....	48
Tabla 8 Prueba de aceptación. Autenticarse. Fuente: Elaboración del Autor	58
Tabla 9 Prueba de aceptación. Gestionar Usuario. Fuente: Elaboración del Autor	59
Tabla 10 Diferencias entre SCRUM y XP. Fuente: Elaboración del Autor.....	70
Tabla 11 Resumen de HU. Fuente: Elaboración del Autor.....	71
Tabla 12 Lista de Chequeo Fuente: Elaboración de la UCI.....	74

Resumen

La necesaria incorporación de herramientas informáticas en el desarrollo de la gestión de los procesos ya sean económicos, políticos o de otras índoles se hace imprescindible para la práctica de la misma. La gestión de los programas de la formación doctoral en Cuba no está exenta de esta realidad. La Universidad de Matanzas y en particular el departamento de Grado Científico hasta este momento no han contado con una plataforma interactiva que permita viabilizar y gestionar el proceso desde que ingresa el aspirante hasta que finalmente concluye su participación en un programa de doctorado. Han sido presentadas diferentes propuestas de software en años anteriores pero no han cumplido con los requisitos necesarios para su implantación por lo que sigue siendo una necesidad vigente. Constituyendo el objetivo principal de esta investigación, realizar un sistema web que gestione los Programas de Formación Doctoral utilizando el framework Symfony3, como metodología, XP y cómo lenguaje de programación, PHP.

Abstract

The necessary incorporation of computer tools in the development of process management whether economic, political or otherwise is essential for the practice of it. The management of doctoral training programs in Cuba is not exempt from this reality. The University of Matanzas and in particular the department of Scientific Degree until now have not had an interactive platform that allows to make viable and manage the process from the moment the candidate enters until finally he concludes his participation in a doctorate program. Different software proposals have been presented in previous years but they have not complied with the necessary requirements for their implementation, so it still a current need. Constitute the main objective of this research, make a web system that manages the Doctoral Training Programs using the Symfony3 framework, as a methodology, XP and as a programming language, PHP.

UNIVERSIDAD DE MATANZAS
Facultad de Ciencias Técnicas
Departamento de Informática

Introducción

El nivel de desarrollo alcanzado en la educación superior en el mundo demanda de la cantidad y calidad de los doctores que se forman y trabajan en las universidades. El porcentaje de profesionales que son doctores es uno de los indicadores que se considera para determinar el grado de excelencia de una universidad, instituto, escuela de altos estudios y centro de investigación. Esto no es casual, ya que el doctorado es un grado científico que certifica que quien lo posee es un profesional altamente calificado en su especialidad y que está capacitado para desarrollar y dirigir investigaciones con un alto grado de independencia y creatividad. Por tanto, "Una tesis doctoral es un estudio escrito sobre una investigación de carácter original efectuada por un estudiante o un escritor con base en un análisis de publicaciones hechas por otros sobre un tema dado. La tesis doctoral permite acceder al grado de Doctor." (Korstanje, 2018)

Los doctorados presuponen el grado más alto de formación académica y constituyen la médula del caudal de Recursos Humanos en Ciencia y Tecnología. La participación en los procesos de doctorado por parte de los recursos humanos los capacita para desarrollar proyectos de investigación complejos y para cumplir una función de liderazgo intelectual que permite irradiar en situaciones concretas, un cúmulo apreciable de conocimientos. De ahí su importancia para el desarrollo de un país (Hernández & Rodríguez, 2014).

Las actuales condiciones en que se desarrolla el mundo contemporáneo apelan una mayor excelencia de la educación y su continuo perfeccionamiento, lo que justifica los numerosos cambios ocurridos en el sistema educativo de diferentes países. En este contexto de cambios, la formación pedagógica de los profesores juega un papel decisivo. El debate en torno a la significación de la preparación investigativa y teórico – metodológica de los claustros docentes para el desarrollo de nuestras sociedades es y será un problema que justifica integrar voluntades y poner en práctica alternativas de colaboración (Málaga, 2017) Como parte del trabajo de perfeccionamiento y obtención de la máxima calidad del doctorado cubano, la Comisión Nacional de Grado Científico (CNGC) y la Junta Nacional de Acreditación (JAN) aprobaron la Resolución No. 8 del Presidente de la Comisión Nacional

de Grados Científicos (Ministerio de Educación Superior) con el objetivo de evaluar y acreditar todos los procesos de formación de doctores de la República de Cuba. Después del ensayo de aplicación en tres instituciones superiores, se aprobó en el 2008 para su aplicación en el sistema de grados científicos de la República de Cuba (Científicos, 2015).

El Ministerio de Educación Superior, en su calidad de órgano rector de la educación de postgrado, elaboró sobre la base de lo establecido en el Reglamento de los grados científicos, el procedimiento para la oficialización de las aspiranturas nacionales en las instituciones autorizadas a desarrollar procesos de obtención de grados científicos por la CNGC (Científicos, 2015). El programa de doctorado estructura el proceso formativo en un sistema que asume la investigación científica como centro, y contempla, además, otras actividades curriculares y académicas. El matriculado en un programa de doctorado desarrolla su tema de investigación vinculado a un proyecto e insertado en un grupo de investigación, bajo la dirección de un tutor y en relación con otros especialistas y participantes del programa, de acuerdo con un plan de formación individual (Superior, 2013) Durante este proceso el aspirante debe realizar diferentes actividades académicas, examen de ingreso, exámenes de mínimo (idioma extranjero, problemas sociales de la ciencia y la tecnología y especialidad), sesiones científicas, producción científica, ejercicio de pre-defensa y acto de defensa con el cual concluye su proceso. A todo esto, se llega a la conclusión, de que el proceso de formación doctoral en Cuba y más específicamente en la Universidad de Matanzas es enormemente complejo y genera una cantidad de documentación inmensa la cual resulta muy difícil de procesar y como es lógico, en muchos casos existe pérdida de la misma. Por tanto, después de entrevistar a varias personas encargadas de realizar este proceso se llegó a la conclusión de que es necesario realizar alguna acción de manera tal que se agilice este proceso. Se encontraron varios sistemas que realizan este proceso, pero al estudiarlos más a fondo que no cumplían con todos los requisitos funcionales que se requieren.

Problema científico: ¿Cómo automatizar la gestión de la información asociada a los programas de formación doctoral desde la CNGC?

Hipótesis: Si se desarrolla un sistema web para automatizar la gestión de la información asociada a los programas de formación doctoral desde la CNGC entonces se logrará agilizarlo en la Universidad de Matanzas.

Objeto de estudio: Gestión de la información asociada a los programas de formación doctoral desde la CNGC.

Campo de Acción: Informatización de la gestión de la información asociada a los programas de formación doctoral desde la CNGC.

Alcance: Está dirigido a la Universidad de Matanzas, aunque el producto final puede ser extendido y utilizado en las demás universidades del país.

Objetivo general: Desarrollar un sistema web para agilizar la gestión de la información asociada a los programas de formación doctoral desde la CNGC.

Objetivos específicos:

1. Elaboración de un Marco Teórico-Referencial sobre la gestión de la información asociada a los programas de formación doctoral desde la CNGC.
2. Desarrollar un sistema web para automatizar la gestión de la información asociada a los programas de formación doctoral desde la CNGC.
3. Validar la solución propuesta de sistema web para automatizar el proceso de la gestión de la información asociada a los programas de formación doctoral desde la CNGC.
4. Implantar el sistema web para automatizar el proceso de la gestión de la información asociada a los programas de formación doctoral desde la CNGC.

Métodos teóricos empleados

Histórico-lógico: Admitió esclarecer la historicidad del problema a solucionar y cómo funciona el ajuste y la solicitud de servicios, su definición, sus características, evolución y desarrollo. Así como los patrones que intervienen en la conformación del mismo.

Analítico-sintético: Permitió realizar el estudio teórico de la investigación facilitando el análisis de documentos y la extracción de los elementos más importantes acerca del proceso de desarrollo del sistema.

Inductivo-deductivo: Implicó la correcta deducción de los casos particulares. A partir de este supuesto se pueden inferir conclusiones y establecer predicciones, de las cuales se deducen las valoraciones de que es necesario desarrollar un sistema web para gestionar los programas de formación doctoral.

Enfoque de sistema: Proporcionó la orientación general del estudio como un maquetado principal que gestione las otras partes del sistema.

Métodos empíricos empleados

Observación: Se observó cómo se llevaba el proceso de los programas de formación doctoral, las deficiencias presentadas durante el mismo, y los pasos que se siguen hasta que concluye el proceso.

Entrevista: permitió realizar el estudio teórico de la investigación facilitando el análisis de documentos y la extracción de los elementos más importantes acerca del proceso de los programas de formación doctoral permitiendo conocer cuáles eran las inquietudes y sugerencias por parte del personal especializado en el tema y qué hacer para facilitar y agilizar su trabajo. Esto permitió a su vez el levantamiento de los nuevos requisitos funcionales y no funcionales para el despliegue de las nuevas funcionalidades.

Análisis documental: Se analizaron todos los documentos relacionados con el proceso de formación doctoral, dígame los creados por la Comisión de Grados de la Universidad de Matanzas, así como las leyes que son a nivel nacional.

El contenido de este trabajo consta de tres capítulos, definidos de la siguiente forma:

Capítulo 1: Fundamentación Teórica. Incluye un estudio del proceso de formación doctoral detalladamente y de los anteriores sistemas elaborados. Se muestra, además, algunos conceptos asociados al dominio del problema, la metodología que guiará el proceso de desarrollo del software, el lenguaje de modelado empleado por la misma, los Lenguajes de Programación Web, el Entorno de Desarrollo Integrado, los Sistemas Gestores de Contenido y de Bases de Datos; así como las herramientas empleadas exponiendo sus principales características y su utilidad para la solución propuesta.

Capítulo 2: Solución Teórica del Problema Científico. Se argumenta la solución que se propone, mostrando una planificación inicial del proyecto, con el empleo de la metodología ágil de desarrollo de software XP. Se desarrolla la propuesta para darle solución a la situación problémica, presentando una planificación por iteraciones. Muestra además los requisitos funcionales y no funcionales con los que deberá cumplir, así como aspectos esenciales del proceso de implementación.

Capítulo 3: Validación de la solución propuesta. Contiene los resultados obtenidos de las pruebas de funcionales y de aceptación realizadas al sistema así como las herramientas utilizadas para la realización de las pruebas de seguridad. Se hace un análisis de los resultados obtenidos, basándose en el criterio de los clientes y los propios de la metodología

de software. Se realiza además un estudio de los beneficios tangibles e intangibles de la realización del software.

Capítulo 1 Fundamentación teórica

1.1 Introducción

Este capítulo contiene los conceptos fundamentales con respecto al dominio del problema, el objeto de estudio, los antecedentes del trabajo y el análisis de algunas de sus características elementales con el objetivo de permitir una mejor orientación para su solución. Se hace referencia además, a la metodología que guiará el proceso de desarrollo del software, las herramientas de Ingeniería del Software Asistida por Computadora, el lenguaje de modelado empleado, los lenguajes de programación, el Entorno de Desarrollo Integrado (IDE), los Sistemas Gestores de Contenido y de Bases de Datos utilizados. Se realizará un estudio detallado de cuáles son las que debido al tiempo y costo conviene ser utilizadas, se tendrá en cuenta cuales son las que los desarrolladores dominan con más fuerzas para que se realice con la calidad requerida y salga satisfecho el cliente.

1.2 Conceptos fundamentales asociados al dominio del problema.

Aspirante: persona que pretende un empleo, distinción o título (Real Academia Española, 2019).

Doctorado: grado máximo académico concedido por la universidad o por determinadas escuelas técnicas superiores, tras realizar y defender públicamente la tesis doctoral. En Latinoamérica, se confiere formalmente el título de doctor a los graduados de un doctorado por investigación, generalmente después de obtener una licenciatura con su tesis y finalmente sustentar la tesis doctoral (Oxford University Press, 2019).

Programa de doctorado: es un proceso de formación científica que se estructura en un sistema de influencias planificadas de carácter científico, docente, metodológico y educativo que tienen a la investigación científica como centro de la formación del aspirante y contempla la participación en cursos, seminarios, talleres de tesis, realización de exámenes, trabajo en grupos y en redes, participación en eventos, publicaciones y otras. En el Programa el aspirante trabaja bajo la dirección de un tutor y en vínculo con otros especialistas y participantes del programa. El Programa incluye el sistema de evaluación del aspirante y culmina con la presentación y defensa de una tesis doctoral ante un tribunal constituido (Comisión Nacional de Grados Científicos, 2005).

Curriculum Vitae: es un concepto latino que significa “carrera de la vida”. Permite referirse al conjunto de experiencias de un sujeto, entre ellas las laborales, las educacionales y las

vivenciales. Es una relación ordenada de los datos académicos, de formación y profesionales de una persona (Definición.de, 2019).

Sistemas web: también conocidos como "aplicaciones Web" son aquellos que están creados e instalados no sobre una plataforma o sistemas operativos. Sino que se alojan en un servidor en Internet o sobre una intranet (red local) (Baez, 2014).

Tesis: Trabajo de investigación por lo general monográfico o investigativo, que consiste en una disertación y comprobación de hipótesis previamente establecidas, para demostrar una capacidad analítica y el manejo de procedimientos de investigación.

Arquitectura cliente-servidor: esta arquitectura se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes. Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un Sistema Gestor de Base de Datos (de aquí en adelante SGBD). Por otro lado, los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red (Álvarez Peña, 2017).

Software libre: el término *software* libre se refiere a un programa de ordenador con libertad para su utilización, distribución, modificación y estudio. Desde el punto de vista técnico-legal, se considera *software* libre a los programas licenciados en términos que garantizan a sus usuarios el derecho de ejecutarlos, copiarlos, distribuirlos, estudiarlos, cambiarlos y mejorarlos (Zanotti, 2016).

Patrones de diseño: Los patrones de diseño son un conjunto de reglas que describen como afrontar tareas y solucionar problemas que surgen durante el desarrollo del software. Estos identifican y especifican abstracciones que van más allá del simple ámbito de clases e instancias, o componentes. Para que una solución sea considerada un patrón debe poseer ciertas características como son efectividad habiendo resuelto problemas similares en ocasiones anteriores y reusabilidad permitiendo su aplicación a diferentes problemas de diseño en distintas circunstancias (García D. C., 2016).

1.3 Objeto de estudio

El objeto de estudio es la gestión de la información asociada a los programas de formación doctoral desde la CNGC, por tanto, surge la necesidad de hacer una descripción más detallada de los elementos que se pretenden transformar y perfeccionar.

1.4 Proceso de formación doctoral

Problemática planteada.

Se presenta la propuesta de tesis de doctorado al departamento al cual pertenece el solicitante. El Departamento emite un documento aprobando o no el tema, entonces, si lo aprueban, pasa el Consejo Científico de la Facultad correspondiente, si no se le envía un documento en el que se le explica los motivos por los cuales no se le aprobó la propuesta planteada. El Consejo Científico dictamina la propuesta de tema para su aprobación por el CITMA y anexará a la documentación establecida un aval con la autorización para trabajar en el tema propuesto. El solicitante hará entonces una presentación del tema a trabajar, en el programa autorizado de la institución dirigido por un departamento y tiene que pertenecer a un proyecto de investigación asociado a una línea de investigación.

El Área Autorizada debe garantizar la participación en esta actividad de al menos 5 doctores del área de conocimiento del tema propuesto. Posterior a la discusión y aprobación o no por el colectivo del Área Autorizada, su jefe elaborará el Dictamen que aparece en los modelos de la CNGC, si no es aprobada la propuesta debe prepararse para una nueva presentación, de no aprobar en la nueva presentación se le desaprueba la propuesta. El solicitante con toda la documentación exigida la presentará al asesor de postgrado de la VRIPG, el cual revisará si la documentación regulada está correctamente elaborada, comunicando al Presidente de la Comisión de Grados de la Universidad que el solicitante puede presentar el tema ante esa Comisión. El Presidente de la Comisión de Grados de la Universidad, fijará la fecha de su presentación ante ese órgano comunicándoselo al solicitante, el cual hará una presentación resumida ante la comisión, si en esta presentación no convence a la Comisión de Grados de la Universidad la misma fijará otra fecha para una nueva presentación, de no aprobar se le desaprobará la propuesta.

Una vez aprobada la propuesta por la Comisión de Grado Científico de la universidad, el Director de Postgrado elaborará el dictamen definitivo y preparará toda la documentación para enviar a la Dirección de Postgrado del MES. La aprobación definitiva por el CITMA será recogida por la VRIPG en el MES, una vez que reciba la comunicación. El original será entregado al proponente en la Oficina de Postgrado de la universidad, dejando copia y registro del mismo. Los que reciban el tema aprobado y vayan a realizar el doctorado a tiempo parcial o completo, procederán, en el término de un mes, a su inscripción en el Área Autorizada donde realizará su doctorado y en la Comisión de Grados Científicos de la Universidad, la realización o convalidación del examen de ingreso, los que vayan a realizar su doctorado por la modalidad libre, no tienen que cumplir lo establecido anteriormente.

Una vez inscrito en la Comisión de Grados de la Universidad, se le otorgará la categoría de ASPIRANTE. Ya aprobado como Aspirante de un Programa Autorizado, el departamento al que está adscrito controlará que cumpla con el plan aprobado para la defensa en el tiempo establecido del doctorado que comienza con el otorgamiento de la condición de aspirante, la VRIPG será contraparte a este control. Si opta por el Programa Curricular Colaborativo debe presentar la solicitud al Comité de Doctorado y este aprueba o no la solicitud. De no ser aprobado se le comunica al aspirante el motivo y debe cambiar de tipo de programa de doctorado. En caso de que lo aprueben matricula en un Programa de Doctorado que contempla la participación en cursos, seminarios, talleres de tesis, realización de exámenes, participación en eventos, publicaciones y otras. El Programa incluye el sistema de evaluación del aspirante y culmina con la presentación y defensa de una tesis doctoral ante un tribunal constituido.

Los Comités de Doctorados son creados por el rector de la Universidad y son integrados por especialistas que posean grado científico y un reconocido prestigio en el área de la ciencia de que se trate. Los Proyectos de Programas elaborados por los Comités de Doctorado serán analizados por la Comisión de Grados Científicos de la Universidad y se propondrán por el Rector a la Comisión Nacional de Grados Científicos para su aprobación. A partir del Acuerdo de la Comisión de Grados de la Universidad que otorga la condición de aspirante se abre el Expediente del Aspirante en la Secretaría General de la Universidad, el cual debe contar con todos los documentos oficiales y acreditativos del cumplimiento de cada requisito necesario para cumplir en el proceso de formación. Para poder presentarse a la pre-defensa y posterior defensa el aspirante deberá aprobar los exámenes de candidato (Idioma extranjero, ciencias sociales y especialidad en cuestión). Si el aspirante obtuvo el título profesor auxiliar en un tiempo cercano a la realización de estos exámenes (2 años) entonces podrá convalidar los exámenes de Idioma y ciencias sociales. Si el aspirante está en el Programa Curricular Colaborativo y los Comités de Doctorado dictaminan que la preparación que ha tenido en los cursos del programa es suficiente puede solicitar a la Comisión Nacional de Grados Científicos la convalidación del examen de especialidad. Con anterioridad al acto de defensa de la tesis, el aspirante realizará un ejercicio de pre-defensa ante un colectivo científico directamente relacionado con la especialidad, compuesto por doctores y profesores titulares, en un número de 5 a 7 o más, donde al menos haya 5 doctores de la especialidad en cuestión.

Este colectivo es el que tendrá derecho a la votación final para decidir si procede o no su defensa. Los Departamentos docentes de las Facultades serán los encargados de convocar

a la pre-defensa. El acto de pre-defensa estará presidido por el Jefe del Departamento Docente encargado. Para poder realizar la pre-defensa el aspirante además de haber vencido los exámenes de candidato, debe presentar un documento firmado por el tutor, donde se exprese que la tesis se encuentra lista para presentarse a este ejercicio y que su contenido se encuentra reflejado adecuadamente en el resumen y con 30 días de antelación a la pre-defensa, dos ejemplares de la tesis y dos del resumen. El Departamento deberá designar dos oponentes que además de estudiar la tesis, tendrán que realizar los señalamientos necesarios sobre las deficiencias de contenido y de forma que pueda contener la misma, estos señalamientos serán recogidos en un informe el cual se adjuntará al acta de pre-defensa. El acta de la pre-defensa además de tener el informe de los oponentes tendrá, la firma de quien la preside, nombre y apellidos de los participantes con derecho al voto y el organismo al que pertenecen, las conclusiones acerca de si la tesis reúne los requisitos para ser presentada al tribunal para su defensa, así como las modificaciones que se le recomiendan a la misma, la votación recibida en el acto y de forma explícita la correspondencia del resumen con el contenido esencial de la tesis.

Si las modificaciones propuestas no son de gran magnitud, su cumplimiento podrá ser comprobado por el propio jefe de departamento. En casos de que las modificaciones sean substanciales se harán las recomendaciones procedentes al aspirante para la reelaboración de la tesis, la que deberá presentarse nuevamente en un acto de pre-defensa. El tiempo que dispondrá el aspirante para esto será fijado por el Consejo Científico y deberá aparecer en el acta de la pre-defensa, como norma no deben ser más de 90 días hábiles. En casos que después de presentada la tesis en una nueva pre-defensa ésta no resulte aprobada, el aspirante causará baja definitiva de la aspirantura. La tesis elevada al tribunal para su defensa se acompañará de un documento del Departamento, donde se exprese que se verificó que el aspirante realizó las modificaciones a la tesis acordada en la pre-defensa y que la misma se encuentra lista para la defensa. Las defensas de las tesis serán frente a tribunales de grado, que podrán ser de dos tipos: Tribunales de tesis, que se constituirán para la evaluación de una tesis específica y los Tribunales permanentes, que se constituirán para la evaluación de todas las tesis que se defiendan en una determinada especialidad o grupo de especialidades.

Los Tribunales permanentes se crearán por acuerdo de la Comisión Nacional de Grados Científicos, la que controlará directamente su funcionamiento a través del Director de la Sección de la rama a que se refiera. Su conformación se establecerá a partir de las proposiciones que realicen las instituciones autorizadas en las convocatorias que se realicen

al efecto, sobre la base del análisis del Curriculum vitae de cada uno de los propuestos. Los tribunales permanentes tendrán un máximo de 30 miembros, expertos de la rama en cuestión o de ramas afines, y contarán con un Presidente, un Secretario y hasta 4 Vicepresidentes, los cuales constituirán el Ejecutivo del tribunal. Todos los miembros deberán tener grados científicos de doctor. Para cada una de las defensas programadas en las especialidades que atiende un tribunal permanente, se constituirá un tribunal específico, conformado por 7 miembros y dos oponentes. Estos se seleccionarán entre los miembros del tribunal permanente más cercanos al área del conocimiento de que se trate y por especialistas invitados de entre los miembros de otros tribunales permanentes. Uno de los especialistas invitados puede no ser miembro de alguno de los tribunales permanentes. En el acto de defensa de la tesis tienen que estar físicamente al menos cinco miembros del tribunal más los dos oponentes. Uno de esos miembros tiene que ser el Presidente designado por el Tribunal Permanente. Los oponentes ejercerán su voto con los restantes miembros del tribunal.

La votación será directa y secreta y se producirá inmediatamente después de concluida la defensa, sin discusión previa. Los tribunales permanentes, en coordinación con la Secretaría de la Comisión Nacional de Grados Científicos, fijarán las fechas en que realizarán las reuniones previas y los plazos de las defensas en el año, con el fin de emitir las correspondientes convocatorias. Los tribunales de tesis estarán compuestos por siete miembros, cuyas especialidades estén directamente relacionadas con el tema de la tesis. Todos los miembros del tribunal deben poseer el grado científico que se defiende, uno de los cuales presidirá el tribunal. Al menos dos de los miembros deben pertenecer a instituciones diferentes de aquella autorizada en la que se defiende la tesis. La solicitud de aprobación del tribunal de tesis a la Comisión Nacional de Grados Científicos se realiza a través del procedimiento siguiente:

a) Una vez conformada la composición del tribunal de tesis por la comisión de grados científicos de la institución autorizada, ésta se elevará a la Comisión Nacional de Grados Científicos para su aprobación.

b) La comisión de grados científicos de la institución autorizada remitirá a los efectos de la aprobación del tribunal de tesis propuesto, los documentos siguientes:

- Comunicación firmada por el presidente de la comisión de grados científicos de la institución autorizada, en la que solicita la aprobación del tribunal propuesto, según el modelo establecido, formando parte integrante de la misma. La solicitud de

aprobación debe presentarse con no menos de cuatro meses de anticipación a la fecha programada para la defensa de la tesis.

- Resumen del trabajo de la tesis, el que tendrá una extensión de no menos de una cuartilla ni más de dos.

- Fundamentación del tribunal y de los oponentes en la que se exprese la vinculación de las especialidades, tanto de los miembros como de los oponentes, con el trabajo de tesis que será objeto de la defensa. Esta fundamentación se realizará por separado para cada uno de los miembros oficiales y adjuntos del tribunal y para los oponentes.

Los documentos que se incluirán en el expediente del aspirante y se enviarán a la Comisión Nacional de Grados Científicos para la aprobación de grado, serán los siguientes:

En el caso del otorgamiento del grado científico de doctor en ciencias de determinada especialidad:

1. Acta de la defensa de la tesis. En el caso de la defensa ante un tribunal de tesis, ésta deber estar acompañada del visto bueno de la comisión de grados científicos de la institución autorizada. (Modelo 1).
2. Planilla de datos personales. (Modelo 2).
3. Una foto.
4. Caracterización del aspirante. (Documento expedido por el centro de procedencia del aspirante, donde se consigne su actividad política y social).
5. Documento sobre el cumplimiento de los requisitos establecidos para el desarrollo de la aspirantura, debidamente firmado y con cuño (Modelo 3).
6. Fotocopia del diploma o certificado de graduado de la educación superior.
7. Acta de la pre-defensa e informe de los oponentes.
8. Avales recibidos de diferentes personalidades o instituciones sobre el resumen y la tesis (cuando proceda).
9. Dictámenes de los oponentes.
10. Opinión del tutor.
11. Síntesis del contenido de la tesis. (Una cuartilla).
12. Un ejemplar de la tesis defendida.
13. Dos resúmenes de la tesis.
14. Relación de los documentos enviados firmada por el secretario de la comisión de grados científicos de la institución autorizada.

En el caso de los expedientes para el grado de doctor en ciencias, se deben incluir los siguientes documentos:

1. Acta de la defensa. (Modelo 1).
2. Planilla de datos personales. (Modelo 2).
3. Una foto.
4. Un ejemplar del trabajo presentado para la defensa del doctorado en ciencias.
5. Dictámenes de los oponentes.
6. Relación de los documentos enviados firmada por el secretario de la comisión de grados científicos de la institución autorizada.
(Comisión Nacional de Grados Científicos, 2005).

1.4.1 Diagrama de Procesos del Negocio.

Las siguientes imágenes muestran el proceso de formación doctoral descrito en la problemática presentada anteriormente. Para la realización del mismo se utilizó un software llamado Bizagi Process Modeler en su versión 2.7.2. ([Véase Anexo 1. Mapa de procesos del negocio](#))

Además se muestra, específicamente, el modulo del sistema a implementar. Para la realización del mismo se utilizó Visual Paradigm for UML 8.0 Enterprise Edition. ([Véase Anexo 2. Mapa de procesos del negocio implementado](#))

1.4.2 Antecedentes del trabajo

Ámbito Nacional:

Se estudió un sistema implantado en nuestra universidad que no contiene ni almacena la información asociada al doctorado curricular además de presentar errores en su diseño. Otro de los sistemas analizados presenta una pobre experiencia de usuario, no corresponde con los requerimientos del cliente, se enfoca en la CNGC dejando desamparada a las instituciones (universidad) que son las de mayor carga de trabajo en el proceso de formación doctoral mientras que ahora es un sistema que puede ser utilizado en todas las universidades e instituciones mejorando la calidad del trabajo y la necesidad de las mismas.

1.5 Metodología de desarrollo, herramientas y tecnologías.

Para desarrollar la investigación, se hace necesario el estudio de las herramientas, tecnologías y metodologías de desarrollo para darle cumplimiento al objetivo general. A continuación se describen aspectos a tener en cuenta.

1.5.1 Metodología de desarrollo

Una metodología de desarrollo de software se puede definir como un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda en la construcción de un software (Pressman, 2013).

Actualmente existen diversas metodologías recogidas en dos grandes grupos, las denominadas metodologías tradicionales o formales y las ágiles o ligeras. El primer grupo se caracteriza por centrarse en llevar una documentación exhaustiva de todo el desarrollo del software y en cumplir con la planificación realizada en la fase inicial del proyecto. Además, de mostrar cierta resistencia a los cambios debido a que, al consumarse uno, se ven afectados varios componentes del proceso de desarrollo del software y también se caracteriza por necesitar un proyecto de gran cantidad de participantes, pues requiere de un equipo de trabajo capaz de administrar un proceso complejo en varias etapas.

Como alternativa a estos inconvenientes surgen las metodologías ágiles, caracterizándose por ser más orientadas al desarrollo de software, con bajos niveles de formalización en la documentación requerida y por ser, a diferencia de las tradicionales, más adaptables a los cambios, requerir de pequeños grupos de trabajo y por ser apropiadas para entornos volátiles.

A partir de lo analizado hasta el momento, teniendo en cuenta que el proyecto de investigación presenta un pequeño grupo de desarrollo y un corto tiempo para su elaboración en el que pueden surgir cambios en el mismo, se decide emplear una metodología ágil.

Dentro de estas sobresalen XP (*Extreme Programming*) y SCRUM, las cuales se caracterizan por dividir el desarrollo de software en pequeñas iteraciones, denominadas Sprint en el caso de SCRUM e Iteración en el caso de XP y por ser el cliente parte del equipo de desarrollo. Sin embargo, a pesar de ser muy similares presentan diferencias sutiles importantes como se muestra en el [Anexo 3. Diferencias entre las metodologías SCRUM y XP.](#)

De forma general, SCRUM se centra en la gestión de proyectos de forma eficiente mientras que XP se enfoca en la forma de desarrollar; consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al cliente, generalmente el usuario final, pues es uno de los requisitos para llegar el éxito del proyecto.

Teniendo en cuenta estos elementos y los antes expuestos, referidos a las características de las metodologías XP y SCRUM se decidió utilizar la metodología de desarrollo de

software XP debido a que se adapta en gran medida tanto al tipo de proyecto a desarrollar como a las condiciones de trabajo.

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, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. 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. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (Beck, Una explicación a la programación extrema: aceptar el cambio , 2014).

Esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita. Por tanto, se debe responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación. El segundo objetivo es potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software (Beck, Patricio Letelier, & Ma, Metodologías ágiles para el desarrollo de software., 2015)

El proyecto es pequeño, no existe un contrato previo especificando tiempo, recursos y alcance. Para el desarrollo del sistema no se dispone de un contrato con un presupuesto ni un alcance previamente definidos. El cliente forma parte del equipo de desarrollo, mediante la aplicación de XP se puede lograr una retroalimentación mayor y lograr un producto que satisfaga sus necesidades, el riesgo de desarrollo es elevado debido al corto tiempo de entrega planteado. XP está diseñada a mitigar los riesgos en proyectos con estas características y existe poca disponibilidad de personal. El sistema debe ser realizado por dos personas solamente, no siendo posible la existencia de muchos roles ni la especialización en un rol específico por parte de los miembros. Uno de los principios básicos de XP es la programación en equipos pequeños (2 a 12 personas) con pocos roles, pudiendo los miembros del equipo intercambiar responsabilidades en un momento determinado.

Consiste en una programación rápida o extrema. Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Está vinculada a cuatro frases, la planificación, el diseño, la codificación y las pruebas.

1.5.2 Tecnologías y Herramientas

Lenguaje de Programación:

Existen, en la actualidad distintos lenguajes de programación para la Web, principalmente se destacan dos grupos, que se diferencian entre sí por el lugar que ocupan en la arquitectura Cliente - Servidor.

El primer grupo está formado por los lenguajes que se ejecutan en el **lado cliente**: HTML, JavaScript y CSS, todos encargados de aportar dinamismo a la aplicación en los navegadores. También existe un segundo grupo de lenguajes que se ejecutan en el **lado servidor** entre ellos se encuentran ASP.NET, PHP, Java y Python, estos se caracterizan por desarrollar la lógica de negocio, además de ser los encargados del acceso a Bases de Datos y el tratamiento de la información. Las características de los lenguajes empleados se muestran a continuación.

Del lado del cliente:

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento, en el proyecto se emplearon los siguientes:

➤ **HTML5**

El HTML5 (Hyper Text Markup Language, versión 5) es la quinta revisión del lenguaje de programación de la World Wide Web, el HTML. Esta nueva versión reemplaza al actual (X)HTML, corrige los problemas que los desarrolladores web encuentran, así como rediseñan el código, la presente versión se actualiza en función de las nuevas necesidades que demanda la web en la actualidad (Cantón , 2015).

Puede ser considerado como piedra angular (Cantón , 2015) de la Web Semántica. Presenta una serie de ventajas frente al HTML tradicional, como la capacidad de ordenar semánticamente el contenido del documento con etiquetas como *nav*, *header*, *section*, *footer*; incrustar directamente elementos multimedia como audios, videos y canvas 2d y 3d; un nuevo grupo de tipos de entrada de datos para formularios con validación sin JavaScript; soporte de etiquetas para manejo de grandes cantidades de datos (*Datagrid*, *Details*, *Menu* y *Command*) que permiten generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido en el cliente, (en dependencia del nivel de implementación de HTML5 del navegador utilizado) (Consortium, 2014).

➤ **CSS (Versión 3)**

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 (Cristian, 2014).

➤ **JavaScript (Versión ECMAScript 6)**

Se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (Pérez, 2018).

➤ **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 (Chaffer, 2014).

➤ **Bootstrap (Versión 4)**

Bootstrap, es un *framework* originalmente creado por Twitter, que permite crear interfaces Web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio Web al tamaño del dispositivo en que se visualice. Esta técnica de diseño y desarrollo se conoce como *responsive design* (en inglés) o diseño adaptativo (García L. C., 2015).

El beneficio de usar *responsive design* en un sitio Web, es principalmente que el sitio Web se adapta automáticamente al dispositivo desde donde se acceda. Lo que se usa con más frecuencia es el uso de Media Queries (consultas), que es un módulo de CSS3 que permite la representación de contenido para adaptarse a condiciones como la resolución de la pantalla y si trabajas las dimensiones de tu contenido en porcentajes, puedes tener una Web

muy fluida capaz de adaptarse a casi cualquier tamaño de forma automática (Carpenter & Bithell, 2015).

➤ ***Admin LTE (Versión 2.4)***

Una popular plantilla de aplicación web de código abierto para cuadros de mandos y paneles de control. Está basado en Bootstrap utilizando todos sus componentes además de rediseñar plugins comúnmente utilizados para crear un diseño coherente a interfaces de usuarios en aplicaciones de back-end. Con su diseño modular proporciona una gran flexibilidad y personalización en la hora de definir el diseño (Almsaeed Studio, 2014).

Del lado Servidor:

➤ ***PHP (Versión 7.x)***

PHP es un lenguaje interpretado de alto nivel, multiplataforma, embebido en páginas HTML (*Hyper Text Markup Language*, Lenguaje para el Formato de Documentos de Hipertexto) ejecutado en el servidor. La característica más potente y destacable de este lenguaje es su soporte para una gran cantidad de bases de datos, soporta además el uso de otros servicios que usen protocolos como IMAP (Protocolo de Acceso a Mensajes Electrónicos, por sus siglas en inglés), SNMP (Protocolo Simple de Administración de Red, por sus siglas en inglés), NNTP (Protocolo para la Transmisión de Noticias en Red, por sus siglas en inglés), POP3 (Protocolo de Oficina de Correo u Oficina Postal), HTTP y derivados. Tiene útiles características de procesamiento de texto, las cuales incluyen las Expresiones Regulares Compatibles de Perl, muchas extensiones, y herramientas para el acceso y análisis de documentos XML (Lenguaje de Marcado Extensible, por sus siglas en inglés). Es potente, de alto rendimiento, de fácil aprendizaje y de escaso consumo de recursos (Puertas Ortega & Orellana Zubieta, 2014).

La mayoría de su sintaxis es similar a C, Java y Perl, con solamente un par de características PHP específicas. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil (Schmid, Aulbach, & Saether Bakken, 2013).

En la actualidad existen varios lenguajes de programación que son empleados en disímiles ámbitos, entre ellos C#, y Java, por mencionar algunos, basada en la experiencia del autor de esta investigación, se decide emplear PHP dada la experiencia en el trabajo con este lenguaje.

Las razones por las cuales decide trabajar en este lenguaje se basan en la curva de aprendizaje del autor que se favorece al conocerlo. En la misma cuerda es necesario

destacar la gran cantidad de funcionalidades a implementar, lo cual se complicaría enormemente con un lenguaje desconocido.

➤ **Symfony (Versión 3.x)**

Es un framework PHP que nos permite muy fácilmente utilizar la arquitectura MVC (Model-View-Controller). Fue diseñado para optimizar el desarrollo de aplicaciones Web, proporcionando herramientas para agilizar aplicaciones complejas y guiando al desarrollador a acostumbrarse al orden y buenas prácticas dentro del proyecto. El concepto de Symfony es no reinventar la rueda, por lo que reutiliza conceptos y desarrollos exitosos de terceros y los integra como librerías para ser utilizados por nosotros. Entre ellos encontramos que integra plenamente uno de los framework ORM más importantes dentro de los existentes para PHP llamado **Doctrine**, el cual es el encargado de la comunicación con la base de datos, permitiendo un control casi total de los datos sin importar si estamos hablando de MySQL, PostgreSQL, SQL server, Oracle, entre otros motores. Otro ejemplo de esto es la inclusión del Motor de plantillas **Twig**, que permite separar el código PHP del HTML permitiendo una amplia gama de posibilidades y por sobre todo un extraordinario orden para nuestro proyecto.

Symfony se seleccionó porque se ajusta a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- 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.
- Está 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.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.
- Una potente línea de comandos que facilitan generación de código ahorrando tiempo de trabajo.

Sistemas de Gestión de Base de Datos (SGBD):

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar la definición de los datos, el mantenimiento de la integridad de éstos dentro de la base de datos, el control de su seguridad y privacidad; así como la manipulación de los mismos (Álvarez, 2015).

➤ **MySQL (Versión 5.5.3)**

Los Sistemas Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. En la actualidad existe una gran variedad de SGBD, tanto de tipo comercial como libre. Después de haber realizado un análisis completo entre los más utilizados y teniendo en cuenta los aspectos más significativos por los cuales se pueden comparar, se llega a la conclusión de que el más idóneo para llevar a cabo los objetivos es el MySQL, pues se permite su libre uso y es utilizable en cualquier plataforma (Casillas , Gilbert, & Pérez, 2014).

Actualmente se desarrolla como software libre, pero con la característica de tener doble licencia: libre y otra para uso comercial. Desde sus inicios el MySQL se ha convertido en el Gestor de Bases de datos de código abierto más popular de Internet, principalmente por su simpleza y robustez, además por 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.

MySQL es confiable, fácil de usar, es multiplataforma, multiusuario y permite elaborar consultas con el robusto SQL. Su principal objetivo de diseño fue la velocidad en la lectura de datos. Otra característica importante es que consume muy pocos recursos, tanto de CPU (unidad central de procesamiento) como de memoria. Posee utilidades de administración como backup, recuperación de errores, entre otras. No tiene límites en el tamaño de los registros. Soporta múltiples motores de almacenamiento.

Object Relational Mapping (ORM)

Object Relational Mapping (ORM) es una técnica de programación para convertir las tablas de una base de datos, en una serie de entidades que simplifiquen las tareas básicas de acceso a los datos para el programador (Fundora, 2014).

Ventajas que posee ORM (Microsoft, 2016):

Persistencia transparente: los objetos del dominio no saben nada acerca de la base de datos donde son persistidos, el framework lo resuelve en forma automática utilizando archivos de mapping expresados en XML.

Soporte de polimorfismo: puede cargar jerarquías de objetos en forma polimórfica.

Soporte de los 3 niveles de mapeo de herencia: mapear toda una jerarquía de clases a una sola tabla, crear una tabla por cada clase concreta o crear una tabla por cada escalón de la jerarquía.

Soporte completo de asociaciones: los *frameworks* de ORM soportan el mapeo de todos los tipos de relaciones que pueden existir en un modelo de objetos del dominio (asociaciones 1...1, 1...N, N...M, unidireccionales y bidireccionales).

Soporte de caching: en el contexto de una transacción, puedo disminuir la cantidad de veces que voy contra la base de datos cacheando en memoria los objetos que son accedidos varias veces.

Soporte de múltiples dialectos SQL: se puede independizar completamente del tipo de base de datos utilizada. La aplicación puede persistir sus datos en SQL Server, en Oracle, en MySQL, etc. simplemente cambiando la configuración correspondiente.

Por los elementos antes expuestos se selecciona el ORM Doctrine, en su versión 2.3, para el mapeo de la base de datos y teniendo en cuenta que proporcionará la reutilización de funcionalidades, así como el acceso y modificación de estos de manera más sencilla y rápida pues, al tenerse que realizar una migración de los datos, Doctrine permitirá gestionar fácilmente los cambios que ocurran.

Servidor web

➤ **HTTP Apache (Versión 2.4)**

Es generalmente reconocido como el servidor web más popular del mundo (servidor HTTP). El servidor Web Apache proporciona una gama completa de características de servidor Web, incluyendo CGI, SSL y dominios virtuales. Apache también es compatible con los módulos de plug-in para la extensibilidad. Apache es el software libre, distribuido por la Apache

Software Foundation que promueve diversas tecnologías Web avanzadas libres y de código abierto.

Se selecciona el Servidor Web Apache 2.4 por ser flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos, por ser multiplataforma y adaptarse a diferentes entornos y necesidades. Además por proporcionar diferentes módulos de apoyo para el desarrollo de módulos específicos y ser extensible, característica que adquiere gracias a ser modular lo que ha permitido el desarrollo de diversas extensiones entre las que sobresale PHP, lenguaje de programación del lado del servidor (del Castillo San Félix, 2014). El servidor HTTP Apache es, de los servidores web, con el que más se ha trabajado y con el que más experiencia se ha adquirido por parte del autor del presente trabajo, además de ser un servidor de código libre. Su robustez y gran lista de prestaciones justifican su selección como servidor en la solución propuesta de la plataforma Web. Además, Apache junto con PHP+Symfony y MySQL forman un buen equipo para servir páginas web con contenido dinámico en las que la velocidad y el número de accesos concurrentes sean algo primordial, como es el caso de la aplicación web que se quiere implementar.

Entorno de desarrollo integrado (IDE)

Un Entorno de Desarrollo Integrado (IDE, *Integrated Development Environment*) es un programa informático compuesto por un conjunto de herramientas de programación (Editorbfb, 2015) que les permite a los programadores escribir, compilar, depurar y ejecutar programas (Netbeans.org., 2014).

➤ **PhpStorm (Versión 8.0.1)**

Como entorno de desarrollo se empleó PhpStorm, un programa que sirve como IDE y que brinda un agradable entorno para programar aplicaciones Web con PHP, tiene un potente depurador integrado. Tiene un excelente balance entre una interfaz con múltiples opciones y un aceptable completamiento de código además de permitir ejecutar comandos de consola.

Esta versión extiende el rango de las herramientas construidas y su soporte para productos de software de terceros. Integra nativamente Vagrant, una herramienta para crear medios ambientes de desarrollos reproducibles. Mediante la creación de scripts en una máquina virtual, Vagrant provee una manera de compartir el ambiente de desarrollo en la computadora, y al mismo tiempo otro desarrollador puede lanzar una máquina idéntica (virtual) desde dentro del propio IDE

1.6 Conclusiones parciales del Capítulo:

El análisis de los principales conceptos asociados al dominio del problema facilitó una mejor comprensión del presente trabajo. Además se estudiaron los fundamentos teóricos acerca del proceso Formación Doctoral expuestos en el decreto ley No.133 sobre Grados Científicos y en particular de los Programas, que son la vía fundamental de preparación de los aspirantes a grados científicos. Por otra parte, la selección de las tecnologías, herramientas y la metodología XP para guiar el proceso de desarrollo de software permite lograr mayor productividad, mayores resultados y garantiza la obtención de un producto de mayor calidad que satisfaga las necesidades del cliente.

Capítulo 2 Solución Teórica del Problema Científico

2.1. Introducción

En este capítulo se muestran las principales características del sistema web definidas a partir del estudio realizado en el Capítulo 1: Fundamentación Teórica. Se exponen además los requisitos funcionales y no funcionales con los que deberá cumplir el sistema, así como particularidades del diseño e implementación del mismo. Apoyándose en la metodología de desarrollo de software Programación Extrema (XP) serán abordados los elementos necesarios para describir la solución propuesta con una mayor comunicación, simplicidad, retroalimentación y reutilización del código de desarrollo y principalmente la satisfacción del cliente.

2.2. Flujo de trabajo de los procesos a informatizar en la entidad correspondiente.

La base del éxito del desarrollo de un proyecto está en gran medida en el valor del equipo de trabajo que lo lleva a cabo. Los integrantes del equipo que realizaron esta investigación fueron:

Tabla 1 Equipo de Trabajo Fuente: Elaboración del Autor.

Miembros	Roles XP
Universidad de Matanzas Sede Camilo Cienfuegos	Cliente
Dcs. Walfredo Glez	Jefe del Proyecto
Lorna Ylenia Blanco González	Programador
Lorna Ylenia Blanco González	Encargado de Pruebas

Para un mejor entendimiento del equipo de trabajo es importante ver como se aplican las prácticas según la metodología XP.

- La presencia y disponibilidad del cliente es importante dentro del equipo, las relaciones con este son muy necesarias pues las personas vinculadas de forma directa con el desarrollo del sistema, en caso de cualquier duda, pueden acceder a él rápidamente, evitando atrasos innecesarios durante el proceso de desarrollo.
- El diseño debe ser simple y los programadores han de seguir un estándar de programación o deben tener el mismo estilo programando. El código debe ser

reestructurado contantemente con el objetivo de remover duplicaciones, 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.

- 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.
- Se deben ir proponiendo pequeñas versiones que irán aumentando en complejidad y dimensión en la medida del tiempo y en caso de existir alguna anomalía se lleva a cabo la reprogramación del código sin modificar su principio funcional, cada segmento de código es integrado en el sistema una vez que esté listo, de esa forma el sistema puede llegar a ser integrado y construido varias veces en un mismo día.
- El tiempo de trabajo por semana no debe exceder las 40 horas, esto contribuye a evitar el cansancio mental de los desarrolladores y con ello disminuye la probabilidad de errores.
- La producción de código está dirigida por las pruebas unitarias las cuales tienen como objetivo principal evitar errores y que estos se detecten lo más rápido posible. Las pruebas se convierten en una herramienta de desarrollo, no un paso de verificación que puede despreciarse si a uno le parece que el código está bien.

2.3. Requisitos del software

Un requisito es la condición o capacidad que un usuario necesita para poder resolver un problema o lograr un objetivo. También se define como la condición o capacidad que debe exhibir o poseer un sistema para satisfacer un contrato, estándar, especificación, u otra documentación formalmente impuesta (Sánchez del Río, 2015).

2.3.1. Requisitos funcionales

Los requisitos funcionales son aquellos que indican lo que debe hacer el producto, son las capacidades con las que debe cumplir el mismo (Sánchez del Río, 2015). En la presente investigación, a partir de entrevistas y reuniones con el cliente, se obtuvieron los siguientes requisitos funcionales.

1. Gestionar Usuarios

1.1. Gestionar Usuario.

1.2. Autenticar

1.3. Registrar

2. Gestionar Institución
 - 2.1. Gestionar Institución.
 - 2.2. Gestionar Tipo de Institución.
 - 2.3. Gestionar Área de Institución.
 - 2.4. Gestionar Tipo de área de institución.
 - 2.5. Gestionar Grupo Institucional.
 - 2.6. Gestionar Tipo de Grupo Institucional.
 - 2.7. Informe de instituciones independientes.
 - 2.8. Informe de instituciones nacionales.
 - 2.9. Informe de instituciones internacionales.
3. Gestionar Programas.
 - 3.1. Gestión de Programas.
 - 3.2. Gestionar tipo de Programa.
 - 3.3. Informe de programas internacionales en Cuba.
 - 3.4. Informe de programas cubanos-internacionales.
 - 3.5. Informe de programas nacionales.
4. Gestionar Convocatoria.
5. Gestionar solicitudes.
 - 5.1. Gestionar solicitudes realizadas.
 - 5.2. Gestionar solicitudes en proceso.
 - 5.3. Gestionar solicitudes finalizadas.
 - 5.4. Gestionar solicitudes canceladas.
6. Gestionar país.
7. Gestionar Categoría Docente.
8. Gestionar Grado Científico.
9. Gestionar Títulos.
10. Gestionar Cargos.

2.3.2. Requisitos no funcionales

Los requisitos no funcionales (RNF) se refieren a las características o cualidades que debe tener el producto para hacerlo atractivo, usable, rápido y confiable. Para el desarrollo del sistema se definieron 28 RNF, los que se clasificaron en RNF de usabilidad, confiabilidad, eficiencia, soporte, restricciones de diseño, de interfaz, seguridad, hardware y software.

1. Requisitos de usabilidad

- **Finalidad:**

RNF1. Este sistema está enfocado a la gestión de la información asociada al proceso de formación doctoral en las diferentes etapas permitiendo mayor organización, control y eficacia tanto para los trabajadores como para los propios aspirantes o solicitantes.

- **Ambiente:**

RNF2. El sistema debe presentar una interfaz de usuario fácil de entender y usar.

RNF3. El tiempo de respuesta brindado por el sistema será menor de 3 segundos.

RNF4. Las computadoras de los clientes solo requerirán de un navegador Google Chrome o Firefox en sus versiones actualizadas.

RNF5. El idioma de todas las interfaces de la aplicación será el español.

RNF6. El sistema mostrará el nombre del usuario que está autenticado en el sistema.

RNF7. El sistema contará con un menú que permite acceder a todas las funcionalidades para entrar los datos y procesar la información.

RNF8. En el sistema no existirán más de 3 interfaces para lograr una funcionalidad completa.

RNF9. Los campos obligatorios serán señalados con un asterisco (*).

2. Confiabilidad

RNF10. El sistema impondrá campos obligatorios para garantizar la integridad de la información que se introduce por el usuario.

RNF11. El sistema no permite la entrada de datos incorrectos.

RNF12. Ante el fallo de una funcionalidad del sistema, el resto de las funcionalidades que no dependen de esta deberán seguir funcionando.

4. Soporte

RNF13. El uso del sistema requerirá un tiempo de preparación previa por los usuarios finales para su correcta explotación.

RNF14. Se deberá recibir entrenamiento para la configuración y el mantenimiento del sistema.

5. Restricciones de diseño

RNF15. La iconografía utilizada será única en cada caso, permitiendo representar todos los conceptos del dominio de la aplicación con un ícono distintivo.

RNF16. Las etiquetas de cada funcionalidad y los campos de cada interfaz tendrán títulos asociados a su función.

RNF17. El sistema presentará los términos capitalizados, es decir, la primera palabra tendrá su primera letra en mayúsculas.

6. Interfaz

RNF18. El sistema presentará una interfaz legible, simple de usar e interactiva.

RNF19. La tipografía y colores serán estándares en toda la aplicación.

9. Seguridad

RNF20. El sistema podrá ser utilizado solamente por usuarios autenticados en el mismo.

RNF21. El sistema mostrará las funcionalidades de acuerdo a quien esté autenticado en el mismo.

RNF22. El sistema deberá garantizar la protección ante acciones no autorizadas.

RNF23. El sistema deberá bloquearse después de cinco intentos de autenticación fallidos.

RNF24. Las contraseñas del sistema deberán tener más de seis caracteres combinando mayúsculas, minúsculas y caracteres especiales.

10. Hardware

RNF25. Los requisitos mínimos de hardware para el servidor:

- ✓ Memoria RAM: 2 GB.
- ✓ Micro procesador: Intel Core i3 o superior.
- ✓ Capacidad Mínima de Disco Duro: 160 GB.

RNF26. Para el Cliente:

- ✓ Procesador: Pentium IV o superior.
- ✓ Memoria RAM: 1GB o superior.

11. Software

RNF27. Para el cliente:

- ✓ Navegador Mozilla Firefox 46.0 o superior.
- ✓ Sistema operativo multiplataforma.

RNF28. Para el servidor:

- ✓ Sistema operativo Linux preferentemente en cualquiera de sus distribuciones.
- ✓ Un servidor Apache 2.4 o superior con módulo PHP 5 o superior disponible

- ✓ Un servidor de base de datos MySQL 5.5.37 o superior.

2.4. Fase de planificación

2.4.1. Historias de Usuario

Las historias de usuarios son la técnica utilizada para especificar los requisitos del software, son lo equivalente a los casos de uso en el proceso unificado y constituyen la base para las pruebas funcionales.

El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario (UH User History en inglés) con el cliente. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. Cuando llega la hora de implementar una UH, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia.

Escala Nominal de Prioridad:

Alta: Se le otorga a las Historias de Usuario que resultan funcionalidades fundamentales en el desarrollo del proyecto, a las que el cliente define como principales para el control integral de proyectos.

Media: Se le otorga 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 son esenciales para el proyecto en desarrollo (Izquierdo, 2014).

Escala Nominal de Riesgo en Desarrollo

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

Media: Cuando pueden aparecer errores en la implementación de la UH 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 (Pressman, 2013).

En la tabla ([Véase Anexo 4. Resumen de HU](#)) se muestran un resumen de todas las Historias de Usuarios (HU) planificadas inicialmente, en las que queda definido el nivel de prioridad con el que deben darle solución a las HU, teniendo en cuenta el riesgo en desarrollo. Se define también la estimación del esfuerzo requerido (en semanas), que no es más que el tiempo en el que se concibió inicialmente el desarrollo de cada HU.

En total se definieron 14 historias de usuarios, a continuación se muestra una de las historias de usuario de mayor complejidad correspondiente al requisito funcional Gestionar Programa de Doctorado que se muestra en la siguiente tabla:

Tabla 2 Historia de Usuario Fuente: Elaboración del autor.

Historia de Usuario	
Número: 4	Nombre Historia: Gestionar Programa.
Usuario: Gestor de Programas	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración Asignada: 2
Programador responsable: Lorna Ylenia Blanco González	
Descripción: Permite mostrar, eliminar, editar y agregar los diferentes Programas.	
Observaciones: Solo usuarios con privilegios correspondientes a estas funcionalidades las pueden ejecutar y se debe verificar la integridad de los datos. Debe existir una jerarquía de acceso a la información lo que conlleva a que un gestor de programa de una institución solo puede gestionar la información correspondiente a la misma.	

2.4.2. Plan de entregas

Las entregas se harán al finalizar cada iteración según los tiempos que se habían precedido con anterioridad para la realización de cada historia de usuario. El cliente tendrá la posibilidad de plantear las pruebas de aceptación con las cuales comprueba que cada una de las historias de usuario ha sido implementada correctamente.

Incidencias

En la planificación de un proyecto es difícil tener en cuenta todos los detalles desde un inicio y en consecuencia ocurren modificaciones al respecto. La poca experiencia del equipo de trabajo en la organización y los contratiempos aparecidos durante su desarrollo inciden en la programación de las iteraciones iniciales y por consiguiente en el plan de las entregas preliminar pactado con el cliente.

Este proyecto no quedó exento ante esta problemática pues sobre la marcha surgieron necesidades de modificar la base de datos en algunas ocasiones, también en el transcurso del proceso de desarrollo de la aplicación hubo variaciones que provocaron la inclusión de otras que no se tomaron en cuenta desde un principio. Demostrando que no es posible una planificación inicial de entregas inalterable. A partir de esto queda aclarado que el cliente es de gran importancia durante todo el proceso para lograr el éxito del proyecto, ya que a través de la comunicación continua se soluciona rápidamente cualquier duda relacionada con los requisitos.

2.4.3. Iteraciones

El proyecto fue dividido en cuatro iteraciones, por lo que se obtuvo un total de cuatro entregas para las cuales se desarrollaron partes de la aplicación completamente funcionales. Para la determinación de cada una de las iteraciones se tuvo en cuenta la opinión del cliente a través de las entrevistas que se le realizaron continuamente antes de comenzar a desarrollar cada iteración donde se tomaron todos los acuerdos necesarios. Una vez concluida la iteración que va a estar conformada por un conjunto de historias de usuarios, teniendo en cuenta los requisitos exigidos por el cliente, la aplicación poseerá mayor número de funcionalidades.

2.4.4. Plan de iteraciones

Teniendo las historias de usuario del sistema definidas y la estimación del esfuerzo para implementar cada una de ellas se procede a realizar la planificación de la etapa de implementación del proyecto. Atendiendo a lo mencionado con anterioridad se decide realizar cuatro iteraciones durante el desarrollo del sistema. A continuación se muestra el plan de duración de las iteraciones ([Véase Anexo 5. Plan de Iteraciones](#)).

2.5. Análisis de los costos.

Cuando se desarrolla un software es imprescindible tener en cuenta los costos que reportará el proyecto, así como un análisis de los beneficios que nos proporcionará.

Principales Conceptos en el Método por Puntos de Función:

- EI: Procesos en los que se introducen datos y que suponen la actualización de cualquier interno.
- EO: Procesos en los que se envía datos al exterior de la aplicación.
- EQ: Procesos consistentes en la combinación de una entrada y una salida en el que la entrada no produce ningún cambio en ningún archivo y la salida no contiene información derivada.

- ILF: Grupos de datos relacionados entre sí, internos al sistema.
- EIF: Grupos de datos que se mantienen externamente.
- PFTe: Total Puntos de Función para las entradas del sistema.
- PFTo: Total Puntos de Función para las salidas del sistema.
- PFTq: Total Puntos de Función para las consultas del sistema.
- PFTif: Total Puntos de Función para los archivos internos del sistema.
- PFTef: Total Puntos de Función para los archivos externos del sistema

Tablas de Valores

EI-EQ

CLASIFICACION DE ENTRADAS Y CONSULTAS	1-4 Atributos	5-15 Atributos	Más de 15 Atributos
0 o 1 ficheros accedidos	BAJA 3	BAJA 3	MEDIA 4
2 ficheros accedidos	BAJA 3	MEDIA 4	ALTA 6
Más de 2 ficheros accedidos	MEDIA 4	ALTA 6	ALTA 6

EO

CLASIFICACION DE SALIDAS	1-5 Atributos	6-19 Atributos	Más de 19 Atributos
0 o 1 ficheros accedidos	BAJA 4	BAJA 4	MEDIA 5
2 o 3 ficheros accedidos	BAJA 4	MEDIA 5	ALTA 7
Más de 3 ficheros accedidos	MEDIA 5	ALTA 7	ALTA 7

ILF

FICHEROS LÓGICOS INTERNOS	1-19 Atributos	20-50 Atributos	Más de 50 Atributos
1 Entidad o registro lógico	BAJA 7	BAJA 7	MEDIA 10
2 - 5 Entidades o registros lógicos	BAJA 7	MEDIA 10	ALTA 15
Más de 5 Entidades o registros lógicos	MEDIA 10	ALTA 15	ALTA 15

EIF

FICHEROS LÓGICOS EXTERNOS	1-19 Atributos	20-50 Atributos	Más de 50 Atributos
1 Entidad o registro lógico	BAJA 5	BAJA 5	MEDIA 7
2 - 5 Entidades o registros lógicos	BAJA 5	MEDIA 7	ALTA 10
Más de 5 Entidades o registros lógicos	MEDIA 7	ALTA 10	ALTA 10

La información que resulte de estas tablas se pondera en una tabla general la cual describe el componente y el nivel en el que se encuentra y se determina el peso de cada componente. Luego se calcula el valor de los puntos de función sin ajustar (PFSA):

$$PFSA = PFTe + PFTo + PFTq + PFTif + PFTef$$

Componente	Bajo	Medio	Alto	Total
EI	8*3=24	6*3=18	2*4=8	PFTe= 50
EO	–	1*4=4	3*5=15	PFTo=19
EQ	8*3= 24	6*4=24	2*6=12	PFTq=60
ILF	–	–	1*15=15	PFTif=15
EIF	–	–	–	PFTef
				PFSA=144

Luego de obtener los puntos de función sin ajustar, debemos calificar cada uno de los factores de valor de ajuste, utilizando una escala del 0 al 5 con el siguiente desglose:

0 - sin influencia

1 - influencia incidental

- 2 - influencia moderada
- 3 - influencia media
- 4 - influencia significativa
- 5 - fuerte influencia en toda la aplicación

Calificamos cada uno de los 14 Ítem y sumamos los grados de influencia (TDI) para obtener el factor de complejidad técnica (FCT):

Ítem	TDI
1. Comunicación de datos	3
2. Proceso distribuido de datos	2
3. Desempeño	3
4. Configuración	2
5. Volumen de transacciones	5
6. Captura de datos en línea	2
7. Eficiencia del usuario final	5
8. Actualización de datos en línea	5
9. Complejidad	4
10. Reusabilidad	2
11. Facilidad de instalación	5
12. Instalación múltiple	2
13. Facilidad de cambio	2
FCT	42

Estimación del software utilizando puntos de función:

PFT_e=50

PFT_o=19

PFT_q=60

PFT_{if}=15

$$PFT_{ef}=0$$

$$PFSA = PFT_e + PFT_o + PFT_q + PFT_{if} + PFT_{ef}$$

$$PFSA = 50 + 19 + 60 + 15 + 0 = 144$$

Puntos de Función Ajustados (PFA)

$$PFA = PFSA \times [0,65 + (0,01 \times FCT)]$$

$$PFA = 144 \times [0,65 + (0,01 \times 42)]$$

$$PFA = 144 \times (0,65 + 0,42)$$

$$PFA = 144 \times 1,07$$

$$PFA = 154,08$$

Líneas de código (LC)

$$LC = PFA \times (\text{Líneas} \times PF)$$

$$LC = 154,08 \times 100$$

$$LC = 15408$$

Esfuerzo [hora/persona] (1 persona trabaja 8h)

$$E = \frac{PFA}{\frac{1}{8} \text{ persona/hora}}$$

$$E = \frac{154,08}{\frac{1}{8} \text{ persona/hora}}$$

$$E = 1232,64 \text{ horas/persona}$$

$$1232,64 \text{ horas/persona} \times 1 \text{ persona} = 1232,64 \text{ horas}$$

Tomando 24 días laborables en el mes y 8 horas productivas al día, obtenemos 192 horas laborables al mes.

Duración del proyecto en meses

$$DM = \frac{1232,64 \text{ horas}}{192 \text{ horas/mes}}$$

$$DM = 6.42 \text{ meses}$$

[A esto se adiciona 1 mes por el margen de error de la estimación por puntos de función y 1 mes para la fase de prueba]

$$DM = 8.42 \text{ meses}$$

Costo total del proyecto

CT = sueldo mensual de 1 persona x cantidad de personas x DM

CT= \$ 500/ persona-mes x 1 persona x 8,42 meses

CT= \$ 4210

2.5.1. Beneficios tangibles e intangibles

La aplicación desarrollada presenta beneficios muy favorables ya que este software le ofrece al personal involucrado una herramienta que facilita su trabajo y da solución a la problemática planteada. El sistema les permite contar con un sistema de fácil acceso y tener un mejor aprovechamiento de las TICs, debido a que gestiona de forma más rápida la información de los Programas de Formación Doctoral, de la Universidad de Matanzas y de esta forma aumenta la calidad del trabajo y por ende disminuye el esfuerzo del usuario para obtener la información necesaria. Con este producto, quedan satisfechas las necesidades del cliente, de manera eficiente.

2.5.2. Análisis de Costos

El desarrollo de un producto informático tiene asociado un costo y el llevarlo a cabo está en dependencia de los beneficios que el mismo pudiese reportar.

De manera que el mayor tiempo posible y los principales esfuerzos en el área estén encaminados a cumplir los objetivos y metas trazadas, además de que al informatizar éste proceso se permite un mayor control y organización sobre los datos. Como se ha planteado anteriormente este sistema informático reporta beneficios considerables, por lo que se puede expresar que es factible.

2.6. Fase de diseño

2.6.1. Arquitectura del sistema

La arquitectura de software: *"es una vista estructural de alto nivel, ocurre muy tempranamente en el ciclo de vida y define los estilos o grupos de estilos adecuados para cumplir con los requerimientos no funcionales"* (UCI, 2015).

➤ Basada en componentes

"El desarrollo de software basado en componentes permite reutilizar piezas de código pre elaborado que permiten realizar diversas tareas, conllevando a diversos beneficios como las

mejoras a la calidad, la reducción del ciclo de desarrollo y el mayor retorno sobre la inversión" (Micrisoft, 2016).

➤ **Modelo Vista Controlador**

"El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo" (Fernández Romero & Díaz González, 2016).

Partes que lo componen (Fernández Romero & Díaz González, 2016):

Modelo: representa la capa de abstracción de los datos, o sea, se manejan los datos y las transformaciones que ocurran con ellos. No cuenta con la responsabilidad de mantener un enlace con las vistas, sino que es tarea del propio sistema notificar a las vistas cuando algo así suceda.

Vista: representa la interfaz con la que interactuará el usuario, visualizando los datos del modelo.

Controlador: maneja las peticiones del usuario actuando sobre los datos y representándolo en la vista.

Ventajas del MVC (Torres Matos & Morales González, 2013):

- *"Una clara separación entre los componentes de un programa; que permite implementarlos por separado".*
- *"Un API (Interfaz de Programación de Aplicaciones) muy bien definido; cualquiera que use el API, podrá reemplazar el modelo, la vista o el controlador, sin aparente dificultad".*
- *"La conexión entre el modelo y sus vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación".*

Patrones de Diseño

Un patrón de diseño es un conjunto de reglas que describen cómo afrontar tareas y solucionar problemas que surgen durante el desarrollo de software. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores.

Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. Los patrones de diseño en su esencia pretenden (Que es un patrón de diseño. [En línea], 2014):

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente. Asimismo, no pretenden:
 - Imponer ciertas alternativas de diseño frente a otras.
 - Eliminar la creatividad inherente al proceso de diseño.

Como se ha mencionado en este trabajo se emplea el *framework* de desarrollo Symfony. Marco de trabajo que trae como ventajas el uso de patrones de diseños que proporcionan una mayor calidad al código y supone una buena práctica para cualquier programador. El autor de la presente investigación emplea los patrones GRASP y GOF en la implementación del sistema.

GRASP

- Experto: crea una capa de abstracción en el modelo, encapsula la lógica de los datos y genera las clases con funcionalidades comunes para las entidades. Por esta vía se crea una relación conteniendo la información de la tabla que representa. Esto es posible con la inclusión de Doctrine para el mapeo de la base de datos.
- Patrón Creador: permite crear objetos de una instancia como vía de acceso a la información del sistema.
- Bajo Acoplamiento: propicia la baja dependencia entre las clases del sistema, esto es apreciable en la no asociación del modelo con la vista o el controlador. Lo que implica un cambio imperceptible ante la necesidad de cambios.
- Alta Cohesión: trabajo por clase asignándole la responsabilidad necesaria a cada una de ellas, por lo que solo es usada para la tarea que fue creada.
- Controlador: todas las peticiones son controladas por un controlador frontal, que es el único punto de entrada a la aplicación.

GOF

- Acción (command): permite la creación de objetos con acciones encapsuladas para no duplicar implementación en presencia de dos eventos que dan respuesta a la misma funcionalidad.
- Decorador (decorator): patrón mediante el cual se pueden añadir funcionalidades dinámicamente, o sea, se modifican, agregan o retiran las propiedades durante su ejecución. Este patrón se evidencia en la clase base.html.twig, padre de todas las vistas, que contiene el Layout de la página. Este archivo, conocido también como plantilla global, guarda el código HTML que es usual en todas las páginas del sistema, para no tener que repetirlo en cada página.
- Instancia única (singleton): garantiza que una clase sólo tenga una única instancia, previendo un punto de acceso global.

2.6.2. Modelado del diseño

Modelo de Datos

Una de las características del sistema a construir es la necesidad de guardar la información para futuras consultas, actividad que solo puede ser realiza mediante la elaboración de una BD. El modelo de datos o diagrama entidad-relación ofrece la información de las tablas más significativas del negocio. A continuación se muestra el diagrama entidad- relación dividido en dos: Gestión de Recursos ([Véase Anexo 6. Diagrama Entidad-Relación: Gestión de Recursos](#)) y Gestión de Procesos ([Véase Anexo 7. Diagrama Entidad-Relación: Gestión de Procesos](#)) uno continuación del otro.

Diagrama de Despliegue

El diagrama de despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado, que muestra las relaciones de la distribución física de los nodos que componen el sistema. Describen la topología del sistema, la estructura de los elementos de hardware y sus relaciones. O sea, se puede decir que se muestra la arquitectura desde el punto de vista del despliegue. Los nodos son recursos de ejecución como los dispositivos de una computadora.

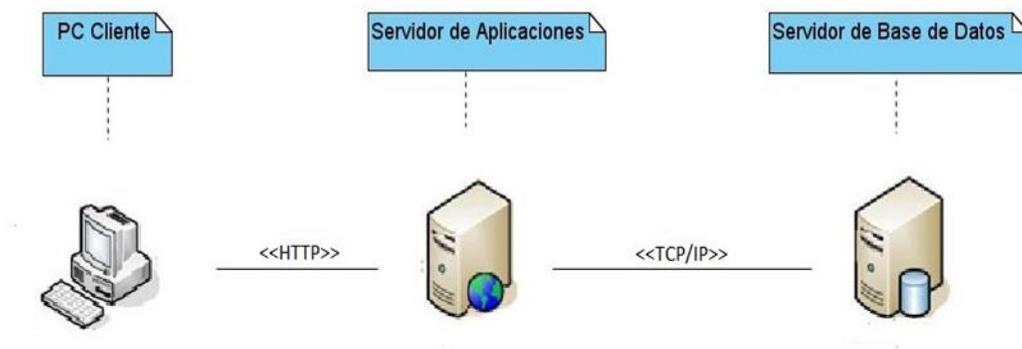


Ilustración 1 Diagrama de despliegue.

2.6.3. Diseño de tarjetas de Clase, Responsabilidad y Colaboración (CRC)

Una de las principales funcionalidades que tiene las tarjetas CRC es mostrar las colaboraciones que se establecen entre las clases que componen el sistema, donde cada tarjeta representa una clase con su nombre en la parte superior. A continuación, se muestra la tarjeta CRC que le corresponde a la historia de usuario Gestionar Programa mencionada anteriormente, el resto de las tarjetas se encuentran en la Documentación del Sistema.

Tabla 3 Tarjeta CRC Fuente: Elaboración del Autor.

Tarjeta CRC	
Clase: Programa	
Superclase :	
Subclase: TipoProgma, Modalidad	
Descripción: En esta clase se guardan los datos relacionados con los Programas.	
Atributos	
id	Int(10)
nombre	Varchar(255)
descripción	Varchar(10000)
es_internacional	Tinyint(1)
aprobadoCNGC	Tinyint(1)
activo	Tinyint(1)
archivo	Varchar(255)

2.7. Implementación

2.7.1. Tareas de ingeniería por historia de usuario

Las tareas de la ingeniería son actividades sencillas que se derivan de las historias de usuario para simplificar la implementación de las mismas, estas se plasman en tarjetas de papel donde se describe lo que se debe realizar. Son muy dinámicas y flexibles ya que pueden ser cambiadas por otras más generales o más específicas, agregarse nuevas o modificarse según las necesidades existentes. El conjunto de tareas de la programación a elegir debe proporcionar suficiente disciplina para alcanzar una alta calidad para el software. Cada una de estas tareas podrá ser comprobada después a través de los casos de prueba (Pressman, 2013).

Debido a la extensión de la cantidad de tareas generadas en el sistema se mostrarán las más esenciales, el resto se podrán encontrar en la documentación del sistema. Los puntos estimados estarán dados por días de trabajo.

Tabla 4 Tarea #24 Insertar Programa. Fuente: Elaboración del Autor.

Tarea	
Número de tarea: 24	Número de Historia de Usuario: 4
Nombre de la Tarea: Insertar Programa de Doctorado	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.33
Programador Responsable: Lorna Ylenia Blanco González	
Descripción: El gestor de programas cuando entra en el sistema selecciona el menú Programas, mostrándose todos los programas existentes y la opción de crear uno nuevo añadiendo: nombre, descripción, si es internacional o no, el archivo que contiene todas las especificaciones del programa, la modalidad y la institución que lo va a desarrollar.	

Tabla 5 Tarea #25 Modificar Programa. Fuente: Elaboración del Autor.

Tarea	
Número de tarea: 25	Número de Historia de Usuario: 4
Nombre de la Tarea: Modificar Programa de Doctorado	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.33
Programador Responsable: Lorna Ylenia Blanco González	

Descripción: El gestor de programas cuando entra en el sistema selecciona el menú Programas, mostrándose todos los programas existentes y la opción de editar donde se muestran los campos para ser modificados.

2.8. Conclusiones del capítulo

En el presente capítulo se elaboraron los elementos descriptivos y artefactos fundamentales correspondientes a las fases de planificación, diseño e implementación que propone la metodología de desarrollo utilizada XP. Se inicia con la conceptualización de los principales objetos del sistema para la realización del diagrama de modelo de dominio y sus principales funcionalidades, para lograr una mejor comprensión de la propuesta. Se logró realizar un levantamiento de los requisitos funcionales mediante la descripción contenida en las historias de usuario. Quedó definida la arquitectura del sistema y durante el proceso de implementación fueron identificados algunos patrones GoF y GRASP que posibilitan la optimización durante esta fase. Se detallaron los planes de entrega e iteración y fueron generadas las tarjetas CRC para establecer el comportamiento y funcionamiento de las clases principales del sistema. Se describe además el diagrama de despliegue, teniendo en cuenta la definición de los requisitos no funcionales identificados.

Capítulo 3. Validación de la solución propuesta.

3.1. Introducción

En este capítulo se realizan las pruebas al software que permiten comprobar la calidad de este producto, lo que constituye uno de los pasos más importantes en el desarrollo de un sistema. No debe existir ninguna característica en el programa que no haya sido probada con la intención de mostrar un error no descubierto hasta entonces y con el fin de verificar la fiabilidad y calidad de la aplicación como un todo. En el siguiente capítulo serán realizadas pruebas funcionales, usabilidad, seguridad y seguridad.

3.2. Pruebas al software

El proceso de pruebas es el instrumento más adecuado para determinar el status de la calidad de un producto. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos o si es el software que se quería desarrollar. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de Prueba (González, 2017).

Una estrategia de pruebas integra los métodos de diseño de los casos de prueba para lograr un software eficaz. La prueba es un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática. Una estrategia de pruebas debe incluir tanto pruebas de alto como de bajo nivel. Son parte de la **Verificación y Validación** incluidas en el aseguramiento de la calidad del software.

Verificación: Comprobar que el software está de acuerdo con su especificación, donde se debe comprobar que satisface tanto los requerimientos funcionales como los no funcionales.

Validación: El objetivo es asegurar que el software satisface las expectativas del cliente.

3.3. Pruebas funcionales

Suelen ser las últimas pruebas que se realizan a la hora de dar el pase a producción a cualquier tipo de software. A este tipo de pruebas se le denomina también de “Caja Negra”, ya que los “tester” se centran en analizar los datos de entrada y salida para definir unos casos de prueba que estarán listos antes del inicio de estas (Testing, 2018).

Plan de Pruebas

Según (Loaiza, 2015) el plan de pruebas de software se elabora con el fin de especificar qué elementos o componentes se van a probar para que el grupo de trabajo pueda realizar el

proceso de Validación y Verificación de los requerimientos funcionales y no funcionales. Además, a través del plan de pruebas se puede continuar con la trazabilidad de los requerimientos, con lo cual el grupo de trabajo, identifica el porcentaje de avance que se ha logrado hasta cierto momento. Al desarrollar el plan de pruebas, se puede obtener información sobre los errores, defectos o fallas que tiene el prototipo, así se realizan las correcciones pertinentes, según el caso y se asegura la calidad del producto que se está entregando al cliente (González, 2017).

En la siguiente tabla se resumen las pruebas que se realizan al sistema web:

Tabla 6 Plan de Pruebas Fuente: Elaboración del Autor.

No.	Historia de Usuario	Pruebas a realizar	Fecha de Inicio	Fecha de fin	Encargado
1	Gestionar Usuario	-Test de insertar usuario. -Test de modificar usuario. -Test de eliminar usuario. -Test de autenticar. -Test de registrar.	01/06/2019	01/06/2019	Lorna
2	Gestionar Institución	-Test de insertar institución. -Test de modificar institución. -Test de eliminar institución. -Test de insertar tipo de institución. -Test de modificar tipo de institución. -Test de eliminar tipo de institución.	01/06/2019	01/06/2019	Lorna

3	Gestionar área de institución.	<ul style="list-style-type: none"> -Test de insertar área de institución. -Test de modificar área de institución. -Test de eliminar área de institución. -Test de insertar tipo de área de institución. -Test de modificar tipo de área de institución. -Test de eliminar tipo de área de institución. 	02/06/2019	02/06/2019	Lorna
3	Gestionar Grupo Institucional	<ul style="list-style-type: none"> -Test de insertar grupo institucional. -Test de modificar grupo institucional. -Test de eliminar grupo institucional. -Test de insertar tipo de grupo institucional. -Test de modificar tipo de grupo institucional. -Test de eliminar tipo de grupo institucional. 	02/06/2019	02/06/2019	Lorna
4	Gestionar Programas.	<ul style="list-style-type: none"> -Test de insertar programa. -Test de modificar programa. 	03/06/2019	03/06/2019	Lorna

		<ul style="list-style-type: none"> -Test de eliminar programa. -Test de insertar tipo de programa. -Test de modificar tipo de programa. -Test de eliminar tipo de programa. 			
6	Gestionar convocatorias	<ul style="list-style-type: none"> -Test de insertar convocatoria. -Test de modificar convocatoria. -Test de eliminar convocatoria. 	03/06/2019	03/06/2019	Lorna
7	Gestionar solicitudes	<ul style="list-style-type: none"> -Test de insertar solicitud. -Test de modificar solicitud -Test de eliminar solicitud. 	04/06/2019	04/06/2019	Lorna
8	Gestionar país	<ul style="list-style-type: none"> -Testo de insertar país -Test de modificar país. -Test de eliminar país 	04/06/2019	04/06/2019	Lorna
9	Gestionar Categoría Docentes.	<ul style="list-style-type: none"> -Test de insertar categoría docente. -Test de modificar categoría docente. -Test de eliminar categoría docente. 	04/06/2019	04/06/2019	Lorna

10	Gestionar Grado Científico.	-Test de insertar grado científico. -Test de modificar grado científico. -Test de eliminar grado científico.	05/06/2019	05/06/2019	Lorna
11	Gestionar Títulos.	-Test de insertar título. -Test de modificar título. -Test de modificar título.	05/06/2019	05/06/2019	Lorna

RF 1- Gestionar Usuarios

RF 1.1- Insertar Usuarios

1- Atributos

- Nombre
- 1er Apellido
- 2do Apellido
- Teléfono
- Contraseña
- Repite la contraseña
- Email
- Tipo de D.I.
- Documento de Identificación
- Foto
- País
- Dirección
- Títulos
- Institución
- Cargos
- Categoría Docente
- Grado Científico
- Rol de Usuario

2- Clases de equivalencia por cada atributo

Tabla 7 Clases de equivalencia. Fuente: Elaboración del Autor.

Atributo	Válida	Representante	Inválida	Representante
Nombre	1- Cualquier combinación de letras.	sfdrgfhjk	2-Vacío	NULL
			3-Insertar caracter especial.	*[¿!"#\$
			4-Insertar números.	123445
1er Apellido	5- Cualquier combinación de letras.	sfdrgfhjk	6- Vacío	NULL
			7-Insertar caracter especial.	*[¿!"#\$
			8-Insertar números.	123445
2do Apellido	9- Cualquier combinación de letras.	sfdrgfhjk	10-Vacío	NULL
			11-Insertar caracter especial.	*[¿!"#\$
			12-Insertar números.	123445
Teléfono	13- Cadena numérica de más de 8 caracteres.	45295613	14-Vacío	NULL
			15- Insertar letras	Dsgfdh
			16- Insertar carácter especial	*[¿!"#\$
			17- Menos de 8 caracteres.	4535
Contraseña	18- Insertar cualquier combinación alfanumérica que contenga mayúsculas, minúsculas y caracteres	Lornayle1*	19- Vacío	NULL
			20- Menos de 8 caracteres.	Lor1*

	especiales con más de 8 caracteres.			
Repite la contraseña	21- Repetir la contraseña anterior.	Lornayle1*	22- Vacío	NULL
			23- Que no sea igual a la contraseña anterior.	Lorjhdsjas1*
Email	24- Cualquier dirección de correo válida.	lornayle@sgd.cu	25- Vacío	NULL
			26- Correo inválido.	Lorjhjbs.cu
Tipo de D.I.	27- Seleccionar en la lista desplegable el valor.	CI Pasaporte		
Documento de Identificación	28- En dependencia de la selección anterior se valida si es numérico para el CI y alfanumérico para el Pasaporte.	CI: 96040110493 Pasaporte: JR54748	29- Vacío	NULL
			30- Insertar letras en el CI.	96gj4676855
			31- Insertar caracter especial en el CI.	96*07263534
			32- Insetar caracter especial en el pasaporte.	JT53627*
Foto	33- Foto con tamaño menor a 2MB y con formato JPG, GIFT o PNG.		34- Tamaño mayor a 2MB	
			35- Formato diferente a JPG, GIFT o PNG.	
País	36- Seleccionar de la lista desplegable.			

Dirección	37- Cualquier combinación alfa numérica.			
Títulos	38- Cualquier combinación de letras o seleccionar de la lista desplegable.		39- Insertar números.	Inge323
			40- Insertar carácter especial	Inge*
Institución	41- Seleccionar de la lista desplegable.			
Cargos	42- Cualquier combinación de letras o seleccionar de la lista desplegable.		43- Insertar números.	Inge323
			44- Insertar carácter especial	Inge*
Categoría Docente	45- Seleccionar de la lista desplegable.			
Grado Científico	46- Seleccionar de la lista desplegable.			
Rol de Usuario	47- Seleccionar de la lista desplegable.			

3- Definir casos de pruebas

No.	Clase Equiv.	Nom b.	1er Apell.	2do Apell.	Teléf.	Contra.	Repite Contra.	Email	D.I	Foto	Direcc.	Título	Cargo
1	1, 5, 9, 13, 18, 21, 24, 27, 28, 33, 36, 37, 38,	Raúl	Gom	Perez	45261530	Raulgom123*	Raulgom123*	raul@sgd.com	96040110493	2mb jpg	Calle 149 entre alvarez y pilar	Ing. Infor mático.	Encar gado de Pruebas.

	41, 42, 45,46, 47.												
2	3, 5,10,15, 18,21,2 5,27,29, 33,36,3 7,39,41, 44,45,4 6,47.	Lo+r	Blanco	NULL	4567fhh7	Raulgom123*	Raulgom123*	NULL	NULL	2mb jpg	Calle 149 entre alvarez y pilar	Ing. 836nd h	Encargado de Pruebas.
3	1, 5, 9, 13, 18, 21, 24, 27, 28, 34, 36, 37, 38, 41, 42, 45,46, 47.	Raúl	Gom	Perez	45261530	Raulgom123*	Raulgom123*	raul@sgd.com	96040110493	3mb	Calle 149 entre alvarez y pilar	Ing. Informático.	Encargado de Pruebas.

4- Tabla de prueba por caso de prueba

No.	1
Requerimiento	Insertar Usuario
Objetivo	Probar la acción de insertar un usuario (clases equivalentes 1, 5, 9, 13, 18, 21, 24, 27, 28, 33, 36, 37, 38, 41, 42, 45,46, 47)
Tipo de Prueba	Funcional
Hardware	Un procesador Pentium IV o superior, disco duro de 500 mb, memoria RAM de 1 GB
Software	Sistema Operativo multiplataforma, Gestor de Base de Datos Wamp o Xamp, Navegador Mozilla Firefox 46 o superior.
Personal	Ingeniero de Pruebas
Casos de Prueba	
Datos de Entrada	Nombre: Raúl 1er Apellido: Gom 2do Apellido: Perez Teléfono: 45261530 Contraseña: Raulgom123* Repite la contraseña: Raulgom123*

	Email: raul@sgd.com Tipo de D.I: C.I Documento de Identificación: 96040110493 Foto: 2mb jpg País: Cuba Dirección: Calle 149 entre alvarez y pilar Títulos: Ing. Informático. Institución: Universidad de Matanzas Cargos: Encargado de Pruebas. Categoría Docente: Doctor Grado Científico: Doctor en Ciencias Informáticas Rol de Usuario: Usuario		
Resultados Esperados	Mensaje: "El usuario se insertó correctamente"		
Resultados Obtenidos	Si(x) No()		
Casos de Excepción:		Comentarios:	
Aprobado por: Lorna Blanco González	Cargo: Líder de pruebas	Fecha: 07/06/2019	

No.	2
Requerimiento	Insertar Usuario
Objetivo	Probar la acción de insertar un usuario (clases equivalentes 3, 5, 10, 15, 18, 21, 25, 27, 29, 33, 36, 37, 39, 41, 44, 45, 46, 47)
Tipo de Prueba	Funcional
Hardware	Un procesador Pentium IV o superior, disco duro de 500 mb, memoria RAM de 1 GB
Software	Sistema Operativo multiplataforma, Gestor de Base de Datos Wamp o Xamp, Navegador Mozilla Firefox 46 o superior.
Personal	Ingeniero de Pruebas
Casos de Prueba	

Datos de Entrada	Nombre: Lo+r 1er Apellido: Blanco 2do Apellido: NULL Teléfono: 4567fhh7 Contraseña: Raulgom123* Repite la contraseña: Raulgom123* Email: NULL Tipo de D.I: C.I Documento de Identificación: NULL Foto: 2mb jpg País: Cuba Dirección: Calle 149 entre alvarez y pilar Títulos: Ing. 836ndh Institución: Universidad de Matanzas Cargos: Encargado de Pruebas. Categoría Docente: Doctor Grado Científico: Doctor en Ciencias Informáticas Rol de Usuario: Usuario
Resultados Esperados	Mensaje: "El usuario no fue insertado, revise los datos "
Resultados Obtenidos	Si(x) No()
Casos de Excepción:	Comentarios:
Aprobado por: Lorna Blanco Cargo: Líder de pruebas Fecha: 07/06/2019 González	

No.	3
Requerimiento	Insertar Usuario
Objetivo	Probar la acción de insertar un usuario (clases equivalentes 1, 5, 9, 13, 18, 21, 24, 27, 28, 34, 36, 37, 38, 41, 42, 45,46, 47)
Tipo de Prueba	Funcional

Hardware	Un procesador Pentium IV o superior, disco duro de 500 mb, memoria RAM de 1 GB	
Software	Sistema Operativo multiplataforma, Gestor de Base de Datos Wamp o Xamp, Navegador Mozilla Firefox 46 o superior.	
Personal	Ingeniero de Pruebas	
Casos de Prueba		
Datos de Entrada	Nombre: Raúl 1er Apellido: Gom 2do Apellido: Perez Teléfono: 45261530 Contraseña: Raulgom123* Repite la contraseña: Raulgom123* Email: raul@sgd.com Tipo de D.I: C.I Documento de Identificación: 96040110493 Foto: 3mb jpg País: Cuba Dirección: Calle 149 entre alvarez y pilar Títulos: Ing. Informático. Institución: Universidad de Matanzas Cargos: Encargado de Pruebas. Categoría Docente: Doctor Grado Científico: Doctor en Ciencias Informáticas Rol de Usuario: Usuario	
Resultados Esperados	Mensaje: "Inserte imagen tamaño máximo 2mb y formato jpg, png o gift"	
Resultados Obtenidos	Si(x) No()	
Casos de Excepción:		Comentarios:
Aprobado por: Lorna Blanco González	Cargo: Líder de pruebas	Fecha: 07/06/2019

3.4. Pruebas de usabilidad

CMMI (*Capability Maturity Model Integration*) es un modelo de mejora del rendimiento de clase mundial para organizaciones competitivas que desean lograr operaciones de alto rendimiento. Está probado en organizaciones y gobiernos a nivel mundial en los últimos 25 años. CMMI consiste en recolectar las mejores prácticas diseñadas para promover los comportamientos que conducen a un mejor desempeño en cualquier organización (Sánchez del Río, 2015).

El propósito de CMMI es el de mejorar los procesos que se realicen en una compañía; adicional a ellos proporciona una guía para el proceso de mejora de la organización, así como su habilidad para manejar el desarrollo, adquisición y mantenimiento de productos o servicios (Sánchez del Río, 2015).

En el siguiente enlace de la página oficial de la UCI de cara a internet está el artículo que hace referencia al proceso de certificación del nivel 2 de CMMI en la UCI:

<http://www.uci.cu/proceso-productivo-de-la-uci-evaluado-con-cmmi-nivel-2>

El uso de las listas de chequeos ha agilizado las pruebas, teniendo en cuenta que constituyen una guía básica y única para el probador en la revisión de los artefactos. Constituyen un apoyo en la ejecución de evaluaciones estáticas garantizando una mayor calidad en los artefactos de apoyo a los sistemas desarrollados por la Universidad de Ciencias Informáticas (UCI), estableciendo un orden de revisión por subtítulos y acápites que coinciden con la organización del artefacto a evaluar.

Esta lista de chequeo cuenta con varios puntos, los cuales serán clasificados antes de ser aplicados, teniendo en cuenta su pertinencia.

Evaluación: Es la forma de evaluar el indicador en cuestión. El mismo se evalúa de 1 en caso de mal (cuando la respuesta al indicador sea “No”) y 0 en caso que elemento revisado no presente errores (cuando la respuesta al indicador sea “Sí”).

NP (No Procede): Se usa para especificar que el indicador a evaluar no se puede aplicar en ese caso.

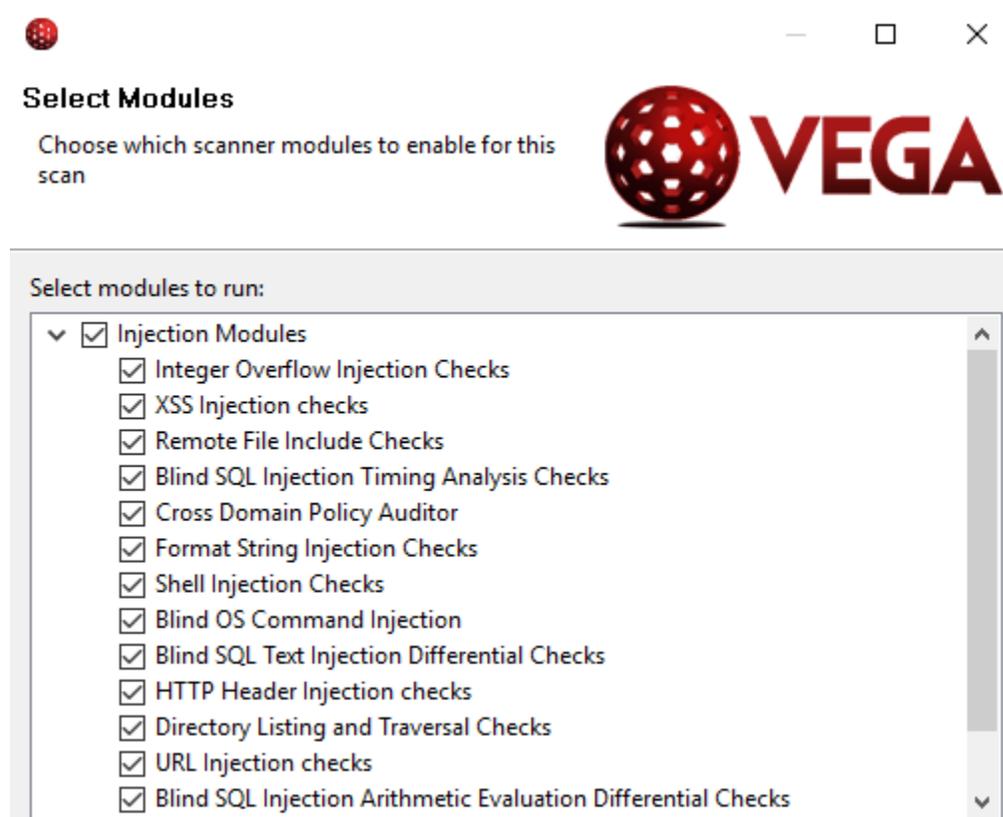
Observación: Especifica los señalamientos o sugerencias que quiera incluir la persona que aplica la lista de chequeo.

(Universidad de Ciencias Informáticas (UCI)).

A continuación se muestra la lista de chequeo con sus respectivas respuestas: [\(Véase Anexo 7. Lista de Chequeo\)](#)

3.5. Pruebas de seguridad

La herramienta utilizada para llevar a cabo las pruebas de seguridad es el software Vega el cual es un escáner de vulnerabilidades de código abierto para probar la seguridad de sitios/aplicaciones web, en la cual nos puede ayudar a encontrar los errores presentados. . Además del escáner posee un proxy para proteger nuestra IP al realizar auditorías, podemos agregar la cookie de una sesión de algún sistema web para que Vega pueda escanear también dentro de las aplicaciones web por vulnerabilidades. Con el uso de este software se obtuvieron los siguientes resultados:



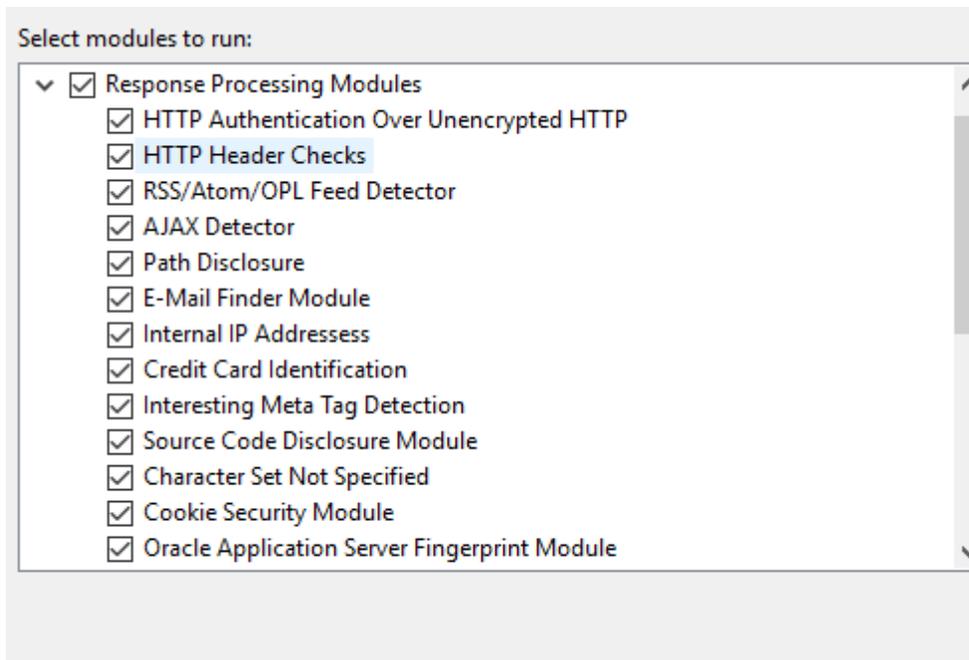


Ilustración 2 Configuración del software Vega

Scan Alert Summary

High		(1 found)
Cleartext Password over HTTP	1	
Medium		(None found)
Low		(3 found)
Form Password Field with Autocomplete Enabled	1	
Internal Addresses Found	2	
Info		(None found)

Ilustración 3 Resultado del software de testeo Vega

Una vez realizadas las pruebas de seguridad, fueron corregidos los errores presentados:

Errores de seguridad alto:

- **Cleartext Password over HTTP:** Este error es porque la prueba fue realizada en un entorno local el cual no tiene instalado un certificado SSL el cual protege los datos que se envían, pero en el entorno real si están instalados el protocolo HTTPS con dichos certificados por lo que no es un problema de seguridad cuando se implante el sistema en la Universidad de Matanzas.

Errores de seguridad bajo:

- **Form Password Field with Autocomplete Enabled:** Este error fue corregido al ponerle al input de la vista login las propiedades AUTOCOMPLETE="off".

3.6. Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra definidas por el cliente para cada HU, y tienen como objetivo asegurar que las funcionalidades del sistema cumplen con lo que se espera de ellas. En efecto, las pruebas de aceptación corresponden a una especie de documento de requerimientos en XP, ya que marcan el camino a seguir en cada iteración, indicándole al equipo de desarrollo hacia donde tiene que ir y en qué puntos o funcionalidades debe poner el mayor esfuerzo y atención (Gutierrez, 2016).

A continuación se muestran unas de las pruebas de aceptación más relevantes:

Prueba de Aceptación al módulo Autenticarse

Tabla 8 Prueba de aceptación. Autenticarse. Fuente: Elaboración del Autor

Pruebas de Aceptación	
Número de Caso de prueba:	Número de historia:
Responsable: Lorna Ylenia Blanco González	
Descripción: Se inserta el usuario y la contraseña para entrar al sistema. Se insertarán de forma incorrecta, dejando campos en blanco para verificar la validación, se tratará de entrar al sistema poniendo una ruta destino para comprobar que no entre solo si se autentifica correctamente. Luego se insertarán los datos de manera correcta para comprobar esta funcionalidad.	
Condiciones de ejecución: El usuario tendrá acceso a las funcionalidades en las que tenga los privilegios correspondientes.	
Entrada/Pasos de ejecución:	

<p>Presionar el botón Entrar con los campos en blanco.</p> <p>Fijar una ruta para acceder a ella sin haberse autenticado.</p> <p>Insertar los datos correctamente y presionar el botón Entrar.</p>
<p>Resultado esperado: El sistema debe alertar al usuario cuando se inserten datos erróneos. Cuando se inserten los datos correctamente, el sistema debe mostrar las opciones a las cuales tiene permiso para acceder el usuario autenticado</p>
<p>Evaluación de la prueba: Satisfactoria.</p>

Prueba de Aceptación al módulo Gestionar Usuario:

Tabla 9 Prueba de aceptación. Gestionar Usuario. Fuente: Elaboración del Autor

Pruebas de Aceptación	
Número de Caso de Prueba:	Número de historia:
Responsable: Lorna Ylenia Blanco González	
<p>Descripción: Se insertan los datos necesarios para crear los usuarios. Se insertarán de forma incorrecta los datos, dejando campos en blanco, se tendrá en cuenta que el usuario no exista en el sistema. Luego se insertarán de manera correcta para comprobar que los datos sean almacenados.</p> <p>Se modifican los datos de los Usuarios. Se modificarán de forma incorrecta, dejando campos en blanco para verificar la validación, luego se modificarán de manera correcta para comprobar que los datos sean almacenados y cargados.</p> <p>Se eliminaran los usuarios.</p>	
<p>Condiciones de ejecución: Los usuarios solo los pueden ser creados, modificados o eliminados, por los usuarios que tenga como rol: Administrador, estos son los únicos que tiene permiso para realizar estas acciones en el sistema. Una persona puede intentar crearse una cuenta de usuario pero esta debe ser aceptada por el Administrador de la CNGC o por el Administrador de la Institución a la que pertenece.</p>	
<p>Entrada/Pasos de ejecución:</p> <p>Dejar campos en blanco.</p> <p>Insertar los datos correctamente.</p> <p>Modificar los datos dejando campos en blanco.</p> <p>Modificar los datos de forma correcta.</p> <p>Verificar que se muestren los nuevos datos.</p> <p>Insertar datos de manera incorrecta.</p>	

Resultado esperado: El sistema debe alertar al usuario cuando se inserten datos erróneos. Cuando se inserten los datos correctamente, el sistema debe almacenarlos en la base de datos y mostrarlos. Cuando se modifiquen los datos correctamente, el sistema debe almacenarlos en la base de datos y mostrar los nuevos.
--

Evaluación de la prueba: Satisfactoria.
--

3.7. Análisis de los resultados obtenidos.

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 encuentran en correspondencia con las especificaciones acordadas y que además cumple con los requerimientos de rendimiento y seguridad. El desarrollo de la aplicación cumple las expectativas trazadas al inicio del proyecto y satisface al cliente en su totalidad.

Este sistema es un gran beneficio para la universidad debido a que le brinda muchas ventajas tanto en tiempo como en costo. Permite una manipulación segura de datos confidenciales, así como hacer consultas dinámicas sobre la información que deseen. También se podrá exportar datos a documentos del formato deseado para ser impresos o enviados por correo. Está realizado para lograr mayor seguridad y confiabilidad en los datos, así como mayor exactitud y organización lo cual estimula el cumplimiento de nuevas estrategias de mejoras.

3.8. Conclusiones del capítulo

Las pruebas realizadas utilizando las técnicas anteriormente planteadas fueron de gran importancia para demostrar el buen funcionamiento del software y el cumplimiento de las peticiones del cliente. Este confirma que el sistema web facilita en gran medida los procesos asociados a los programas de formación doctoral y agiliza así el trabajo de los Especialistas de la Comisión de Grados de la Universidad, por otra parte, queda anexado en la documentación del sistema el Manual de Usuario donde se explica claramente cada una de las funcionalidades de la aplicación.

La implementación del software y la aplicación de las pruebas de validación con resultados satisfactorios demostraron que el software elaborado cumple con los requerimientos especificados por el cliente.

Conclusiones Generales

1. Se sistematizaron los fundamentos teóricos que permitieron desarrollar un sistema web para automatizar los procesos asociados a los programas de Formación Doctoral.
2. Se determinaron las herramientas y la metodología para el desarrollo del sistema web propuesto asumiendo la necesidad de utilizar herramientas libres que facilitaron el trabajo del diplomante para realizar el proceso con la mayor calidad posible.
3. Se implementó el sistema web para automatizar los procesos asociados a los programas de Formación Doctoral de la Universidad de Matanzas o a cualquier universidad del país; asumiendo las acciones y procedimientos de la metodología XP.
3. Se validó la aplicación en dos partes bien diferenciadas, en un primer momento realizando las pruebas tradicionales y utilizando herramientas automatizadas para garantizar la seguridad del sistema y en un segundo momento realizando pruebas de aceptación con el cliente.

Recomendaciones

Luego de analizar los resultados obtenidos de esta investigación se recomienda para su continuidad agregar nuevos módulos hasta lograr informatizar el gran complejo Proceso de Formación Doctoral.

Además fue recomendado por el Departamento de Grados Científicos de la Universidad de Matanzas el agrego de otros datos personales del aspirante que son necesarios para la filtración de información permitiendo poder hacer análisis estadísticos del propio proceso.

Bibliografía Referenciada

- Almsaeed Studio. (2014). *AdminLTE*. Obtenido de AdminLTE: <https://adminlte.io/blog/adminlte-v2.4-release-notes-and-upgrade-guide>
- Álvarez Peña, M. (2017). *Arquitectura cliente-servidor*. [En línea]. Obtenido de <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>
- Álvarez, S. (2015). Sistemas gestores de bases de datos. Introducción a este concepto y características especiales. Obtenido de <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>
- Baez, S. (20 de 10 de 2014). *KnowDo*. Obtenido de KnowDo: <http://www.knowdo.org/knowledge/39-sistemas-web>
- Beck, K. (2014). Una explicación a la programación extrema: aceptar el cambio . Obtenido de <http://metodologiasagiles.com>.
- Beck, K., Patricio Letelier, & Ma, C. (2015). *Metodologías ágiles para el desarrollo de software*.
- Cantón , A. (2015). *Manual de HTML5 en español*.
- Carpenter, J., & Bithell, J. (2015). *Bootstrap confidence intervals: when, which, what?. A practical guide for medical statisticians*. Statistics in medicine.
- Casillas , L., Gilbert, M., & Pérez, O. (2014). *Base de Datos MySQL*. [En Línea].
- Chaffer, J. (2014). *Learning JQuery 1.3: Better Interaction and Web Development with 21 Simple JavaScript Techniques*.
- Científicos. (2015). "Normas y resoluciones vigentes para el desarrollo de los Grados Científicos en la Republica de Cuba". Comisión Nacional de Grados.
- Comisión Nacional de Grados Científicos. (2005). *NORMAS Y RESOLUCIONES VIGENTES PARA EL DESARROLLO DE LOS GRADOS CIENTIFICOS EN LA REPUBLICA DE CUBA*. Ciudad de la Habana.
- Consortium, W. W. (2014). *Borrador actual de especificaciones de HTML5*. [En línea]. Obtenido de <http://dev.w3.org/html5/spec/Overview>.
- Cristian. (2014). *Ventajas de usar CSS*. [En línea].
- Definición.de. (2019). *Definición.de*. Obtenido de Definición.de: <https://definicion.de/curriculum/>
- del Castillo San Félix, A. (2014). *El servidor de web Apache: Introducción práctica: Apache 1.x y 2.0 alpha*. [En línea]. Obtenido de <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/x31.html>
- Editorbfb. (2015). *Qué es un entorno de desarrollo integrado, IDE. Programacion Desarrollo*. [En línea]. Obtenido de <http://programaciondesarrollo.es/que-es-un-entorno-de-desarrollo-integrado-ide/>.

- Fernández Romero, Y., & Díaz González, Y. (2016). Patrón Modelo-Vista-Controlador. *Telem@tica*, 57.
- Fundora, L. R. (2014). *Herramienta web para la gestión de compra de materias*. Cuba: Universidad de Matanzas.
- García, D. C. (2016). *Aplicación informática para gestión de reservas y estancias Hostal Azul Mtzas*. Matanzas.
- García, L. C. (2015). *Portal Web educativo para el desarrollo de habilidades comunicativas en el idioma inglés*. La Habana, Cuba: Universidad de Ciencias Informáticas.
- González, Y. R. (2017). *Aplicación web para mejorar la eficiencia en la planificación de la gestión de los equipos de la reserva estatal en la División de la Empresa de Atención a Equipos Matanzas*.
- Gutierrez, J. (2016). *Pruebas del sistema en programación*. Sevilla.
- Hernández, H., & Rodríguez, P. (2014). "Política y estrategia para la formación doctoral y posdoctoral en el sistema nacional de educación en Cuba".
- Izquierdo, L. R. (2014). *Aplicación web para la evaluación y control de la gestión tecnológica e innovadora en las empresas cubanas*. Universidad de Matanzas.
- Korstanje, M. (2018). "Tesis doctorales, que son y para que sirven?" (Septiembre ed.). Atlante. Cuadernos de educación y desarrollo.
- Loaiza. (2015). *Plan de Pruebas de Software*.
- Málaga. (2017). "El programa de doctorado en ciencias pedagógicas del IPLAC en Venezuela".
- Micrisoft. (2016). *Developer Network [En línea]*. Obtenido de <https://msdn.microsoft.com/es-es/library/bb972268.aspx>
- Microsoft. (2016). *Microsoft Corporation. [En línea]*. Obtenido de <http://www.microsoft.com>
- Netbeans.org. (2014). *Bienvenido a NetBeans y www.netbeans.org. NetBeans [En línea]*. Obtenido de http://netbeans.org/index_es.html.
- Oxford University Press. (2019). *Spanish Oxford living dictionaries*. Obtenido de Spanish Oxford living dictionaries: <https://es.oxforddictionaries.com/definicion/doctorado>
- Pérez, J. E. (2018). *Introducción a JavaScript*.
- Pressman, R. (2013). *Ingeniería de Software. Un enfoque práctico*.
- Puertas Ortega, J., & Orellana Zubieta, F. J. (2014). *Scribd.com*. doi:51830143
- Que es un patrón de diseño. [En línea]*. (2014). Obtenido de <http://www.patrongdiseño.html.com>.
- Real Academia Española. (2019). *DLE*. Obtenido de DLE: <https://dle.rae.es/?id=44JDcIT>

- Sánchez del Río, J. (25 de Abril de 2015). *CANTABRIA*. Obtenido de CANTABRIA:
<http://www.cantabriatic.com/que-es-cmmi-y-para-que-sirve/>
- Schmid, E., Aulbach, A., & Saether Bakken, S. (2013). *PHP Manual, 2*. s. l.: iUniverse Inc. doi:978-0595132287
- Sommerville, I. (2015). *Ingeniería del software*. Madrid, España: Pearson Educación. S.A.
- Superior, C. d. (2013). *"Programa de Doctorado"*.
- Testing, G. (2018). *Globe*. Recuperado el 6 de 4 de 2019, de Globe: globetesting.com
- Torres Matos, Y., & Morales González, Y. (2013). *Plataforma de Gestión de Hardware y Software. Módulo: Mapa Tecnológico*. (s.n., Ed.) Habana.
- UCI. (25 de 07 de 2015). *Entorno Visual de aprendizaje [En línea]*. Obtenido de eva.uci.cu.
- Universidad de Ciencias Informáticas (UCI). (s.f.). *Lista de Chequeo Usabilidad de Sitios Web*. UCI, Departamento de Calidad., La Habana. Recuperado el 5 de junio de 2017
- Valderrama, S. L. (2013). *Programación extrema en pocos minutos: planificando la transición*. Cuba: Tono. Revista Técnica de la Empresa de Telecomunicaciones de Cuba, S.A.
- Zanotti, A. (2016). *El software libre su difusión en Argentina: mercado, estado, sociedad*.

Anexos

Anexo 1. Mapa de procesos del negocio

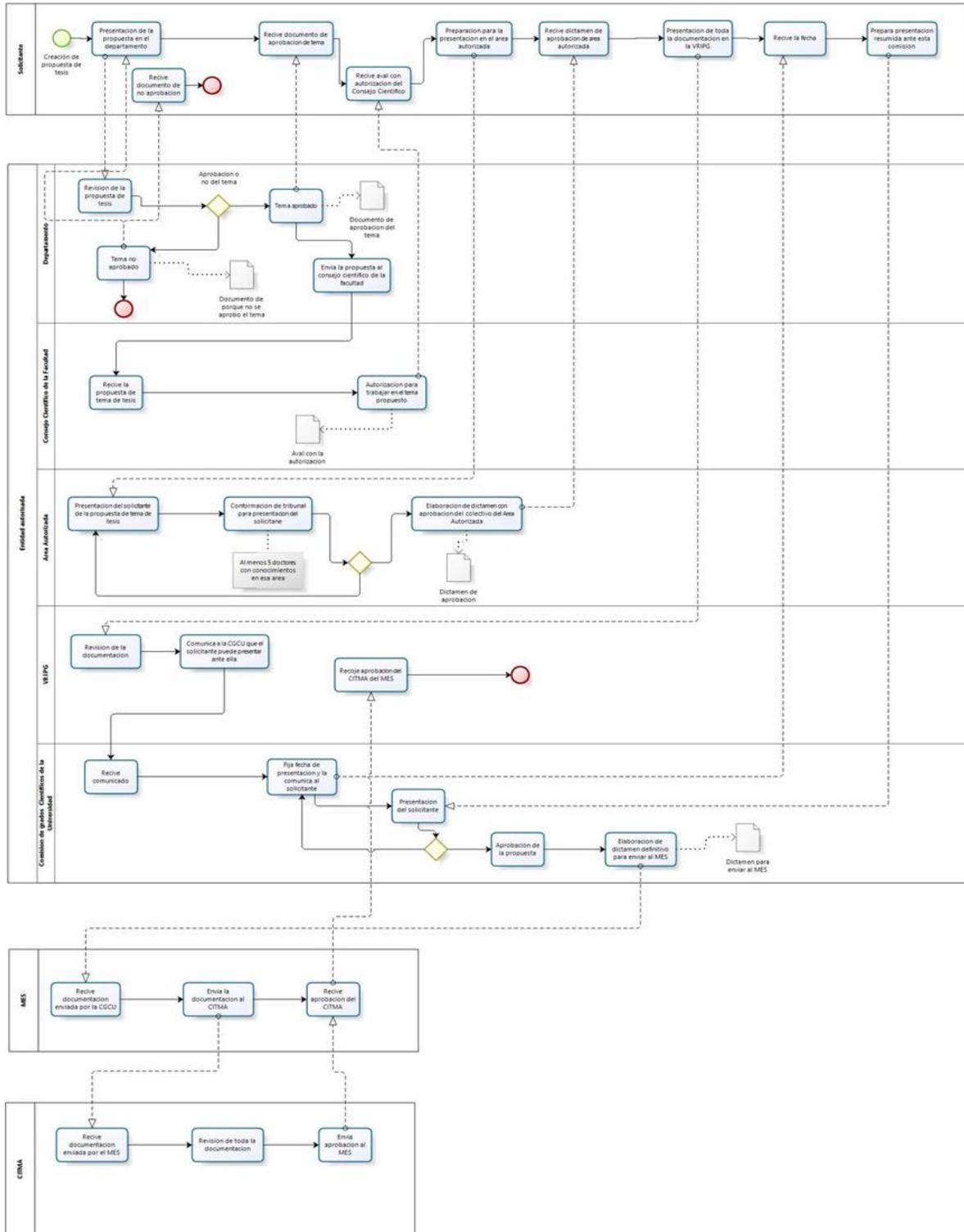


Ilustración 4 Mapa de Procesos. Fuente: Elaboración del Autor.

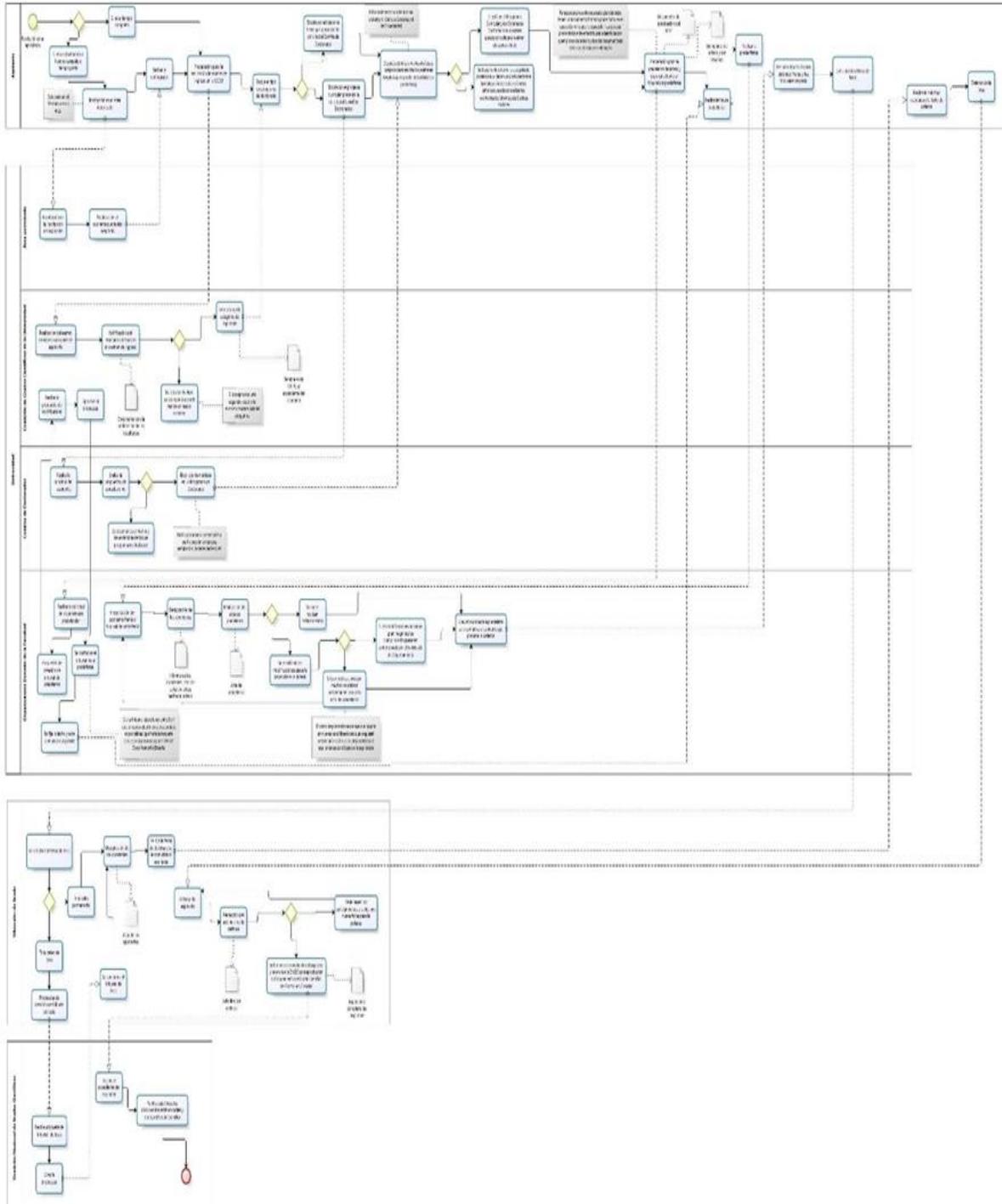


Ilustración 5 Mapa de Procesos II. Fuente: Elaboración del Autor.

Anexo 2. Mapa de procesos del negocio implementado

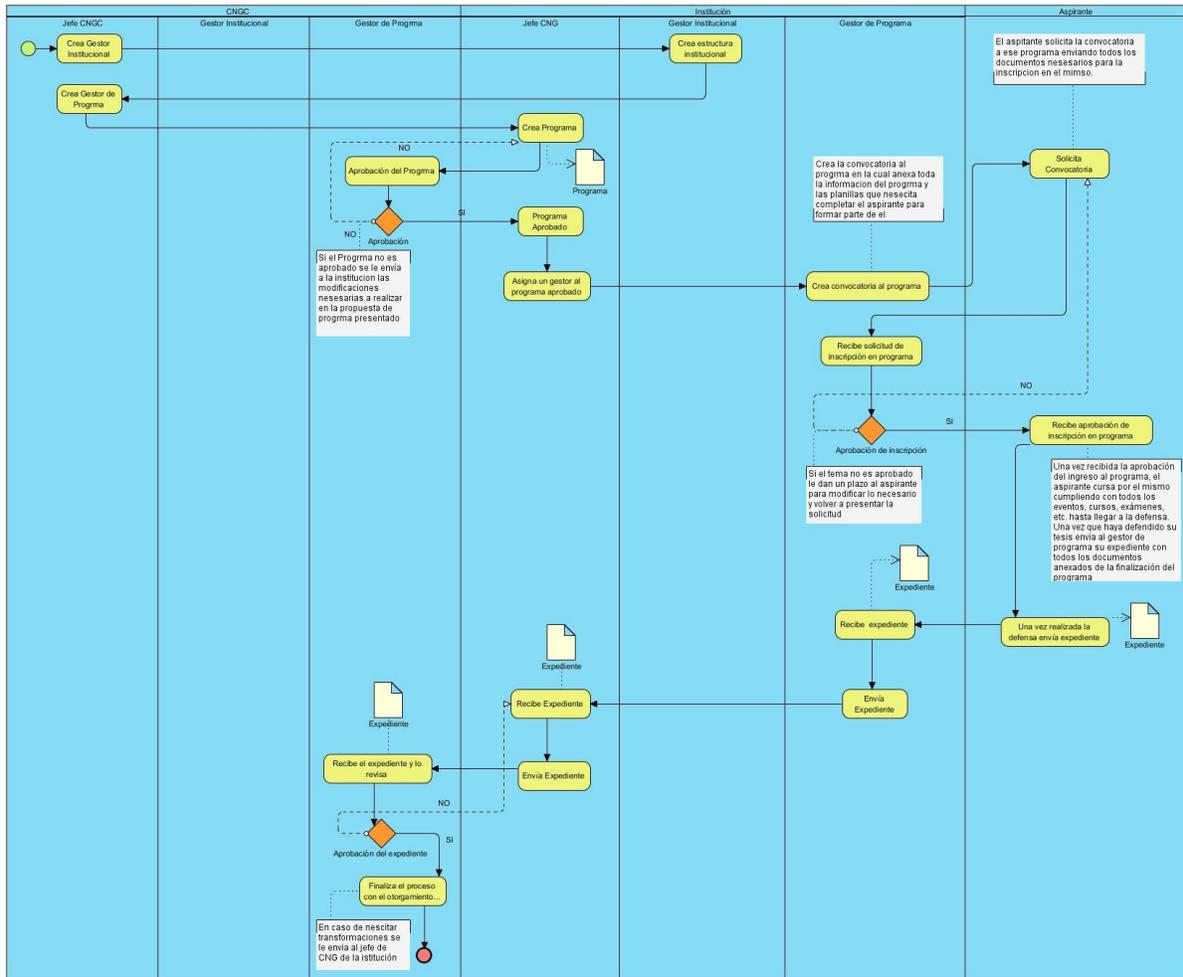


Ilustración 6 Mapa de procesos del negocio implementado

Anexo 3. Diferencias entre las metodologías SCRUM y XP.

Tabla 10 Diferencias entre SCRUM y XP. Fuente: Elaboración del Autor.

Aspectos	SCRUM	XP
Duración de una iteración	Una semana o un mes.	“Aproximadamente 2 meses.” (Sommerville, 2015)
Cambios en las iteraciones	Los equipos no permiten cambios en sus Sprint. Una vez que la reunión de planificación de iteración se ha completado y se ha contraído un compromiso con la entrega de un conjunto de elementos del <i>backlog</i> (lista de tareas identificadas por el equipo SCRUM durante la planificación del Sprint) del producto, estos se mantienen sin cambios hasta el final del Sprint.	Los equipos son más susceptibles al cambio dentro de sus iteraciones, siempre y cuando el equipo no ha empezado a trabajar en una característica particular.
Prioridad de desarrollo	El propietario del producto SCRUM (cliente) prioriza la acumulación de productos, pero es el equipo quien determina la secuencia en la que se desarrollarán los elementos del <i>backlog</i> .	Los equipos trabajan en un orden de prioridad estricta determinado por el cliente.
Prácticas de Ingeniería.	No prescribe prácticas de ingeniería.	Establece prácticas de ingeniería como el desarrollo basado en pruebas, el enfoque en pruebas automatizadas, la programación en parejas, diseño simple y refactorización (Valderrama, 2013)

Anexo 4. Resumen de HU

Tabla 11 Resumen de HU. Fuente: Elaboración del Autor.

No.	Historia de usuario	Tareas asignadas
1	Gestionar Usuarios	1- Insertar Usuario 2- Modificar Usuario 3- Eliminar Usuario 4- Autenticar 5- Registrar
2	Gestionar Institución	6- Insertar Institución 7- Modificar Institución 8- Eliminar Institución 9- Insertar Tipo de Institución 10- Modificar Tipo de Institución 11- Eliminar Tipo de Institución 12- Insertar Área de Institución 13- Modificar Área de Institución 14- Eliminar Área de Institución 15- Insertar Tipo de área de institución. 16- Modificar Tipo de área de institución. 17- Eliminar Tipo de área de institución.
3	Gestionar Grupo Institucional	18- Insertar Grupo Institucional 19- Modificar Grupo Institucional 20- Eliminar Grupo Institucional 21- Insertar Tipo de Grupo Institucional 22- Modificar Tipo de Grupo Institucional 23- Eliminar Tipo de Grupo Institucional

4	Gestionar Programa	24- Insertar Programa. 25- Modificar Programa. 26- Eliminar Programa. 27- Insertar Tipo de Programa. 28- Modificar Tipo de Programa. 29- Eliminar Tipo de Programa.
5	Gestionar Convocatoria	30- Insertar Convocatoria 31- Modificar Convocatoria. 32- Eliminar Convocatoria.
6	Gestionar Solicitud Realizada	33- Insertar Solicitud Realizada. 34- Eliminar Solicitud Realizada.
7	Gestionar Solicitud en Proceso	35- Insertar Solicitud en Proceso. 36- Eliminar Solicitud en Proceso.
8	Gestionar Solicitud Finalizada	37- Insertar Solicitud Finalizada. 38- Eliminar Solicitud Finalizada.
9	Gestionar Solicitud Cancelada	39- Insertar Solicitud Cancelada. 40- Eliminar Solicitud Cancelada.
10	Gestionar País	41- Insertar País. 42- Modificar País. 43- Eliminar País.
11	Gestionar Categoría Docente	44- Insertar Categoría Docente 45- Modificar Categoría Docente 46- Eliminar Categoría Docente
12	Gestionar Grado Científico	47- Insertar Grado Científico. 48- Modificar Grado Científico 49- Eliminar Grado Científico
13	Gestionar Título	50- Insertar Título. 51- Modificar Título 52- Eliminar Título
14	Gestionar Cargo	53- Insertar Cargo 54- Modificar Cargo 55- Eliminar Cargo

Anexo 5. Plan de Iteraciones



Ilustración 7 Plan de iteraciones. Fuente: Elaboración del Autor.

Anexo 6. Diagrama Entidad-Relación.

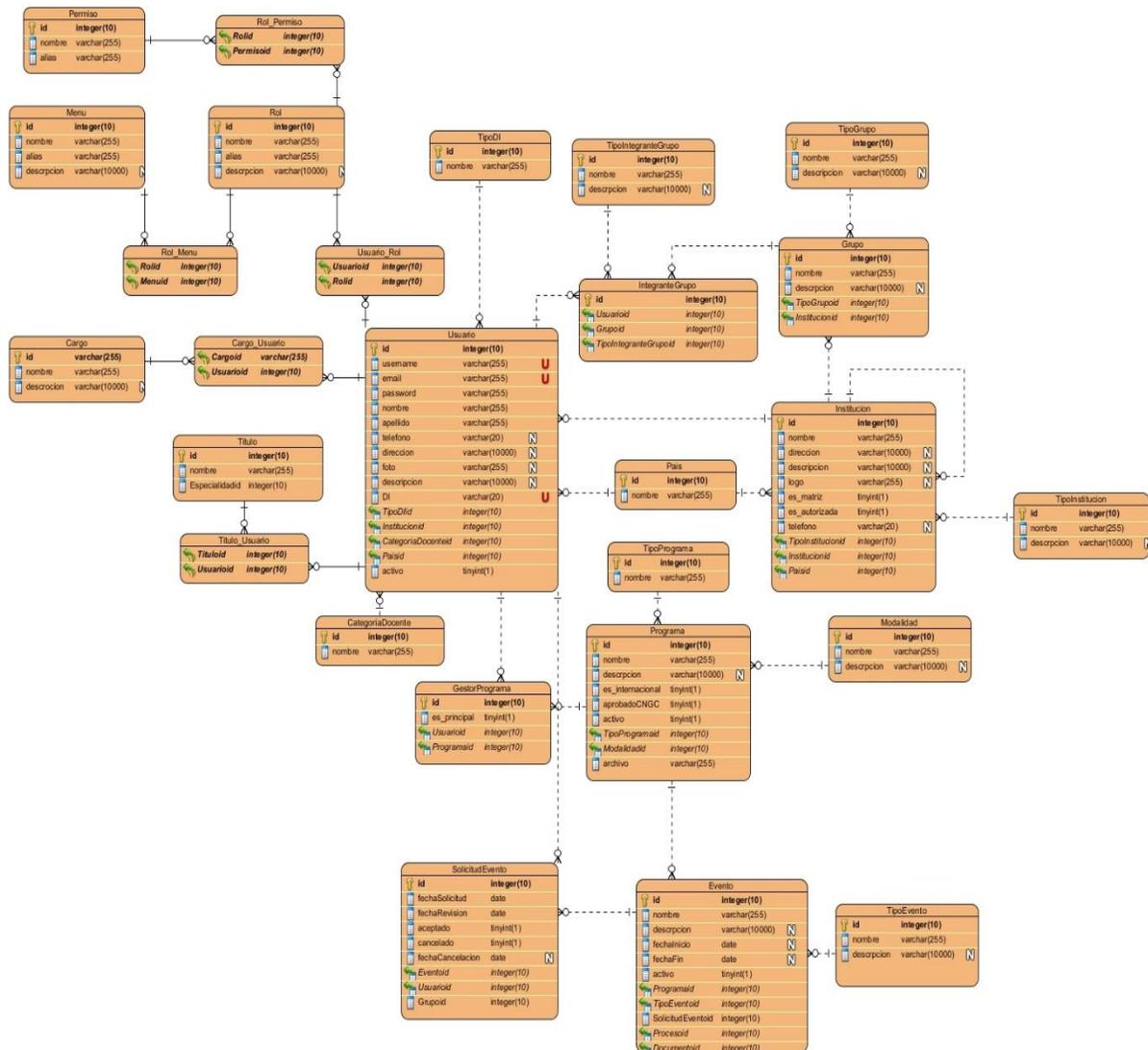


Ilustración 8 Diagrama Entidad-Relación. Fuente: Elaboración del Autor.

Anexo 7. Lista de Chequeo

Tabla 12 Lista de Chequeo Fuente: Elaboración de la UCI.



Elementos definidos por la metodología				
No.	Indicador a evaluar	Evaluación	NP	Observación
Visibilidad del Sistema				
1	¿La página refleja la identidad de la empresa logos, compañía...?		x	
2	¿Cada pantalla empieza con un título que describe su contenido?	0		
3	¿Cuándo se selecciona un icono se diferencia de los no seleccionados?	0		
4	¿Los enlaces del menú se resaltan cuando se seleccionan?	0		
5	¿Los iconos que aparecen se identifican claramente con lo que representan?	0		
6	¿El menú de navegación aparece en un lugar destacado?	0		
7	¿No utiliza más de siete opciones principales en el menú de navegación?	0		
8	¿Si la respuesta a una acción se retrasa, aparece un mensaje o indicio como que el sistema está procesando la acción?	1		
9	¿El sitio le indica al usuario en que parte de la estructura del sitio web se encuentra, es decir si muestra 'Listado de Usuario'?	0		
10	¿El nombre de los enlaces es el mismo que	0		

	el título de la página a la que dirige?			
11	¿El logo de la organización está ubicado en el mismo lugar en todas las páginas, y hacer click en el logo retorna al usuario a la página más lógica (Ejemplo: la página de inicio)?		x	
12	¿Los títulos de las páginas, tablas e imágenes son descriptivos y distintivos?	0		
13	¿Las etiquetas de las categorías describen con precisión la información de las mismas?	0		
14	¿Cuándo una tarea involucra documentos fuente, la interfaz es compatible con las características del documento fuente?	0		
15	¿Las imágenes se muestren con buena resolución?		x	
16	¿No se muestran errores ortográficos?	0		
17	¿No hay ninguna imagen con información relevante?		x	
Lenguaje común entre sistema y usuario				
18	¿El lenguaje es simple, con un tono adecuado?	0		
19	¿La información que se presenta en la aplicación	0		

	es fácil de entender y memorizar?			
20	¿Utiliza los conceptos establecidos para las funciones estándar? ("buscar" para las búsquedas, etc.)	0		
21	¿Evita el lenguaje técnico: términos informáticos o propios de Internet?	0		
22	¿Se utiliza siempre la misma nomenclatura para las mismas funciones?	0		
23	¿Los acrónimos y abreviaturas son definidos al ser usados por primera vez?		x	
24	¿No hace uso de términos extranjeros?	0		
25	¿Utiliza un texto específico y descriptivo en los vínculos?	0		
26	¿La información es de rápida lectura, y con una disposición asequible?	0		
27	¿Los vínculos basados en nombres de la gente, conducen a las biografías cortas o a sus propios blogs, no a un correo electrónico?		x	
28	¿Si se desea incluir un enlace de correo electrónico, se muestra el correo y no el nombre de la persona?		x	
Libertad y control por parte del usuario				

29	¿Existe una manera lógica de acceder a páginas relacionadas o a otras secciones?		x	
30	¿Tras una acción relevante hay una opción de vuelta atrás?	0		
31	¿Si una acción tiene consecuencias, el sistema proporciona información y pide confirmación antes de continuar?	0		
32	¿En las páginas internas hay un acceso a la página de inicio en una zona visible y reconocible?	0		
33	¿El Sitio cuenta con un mapa o buscador que facilite el acceso directo a los contenidos?	0		
34	¿Al dar click en el botón "Atrás" siempre lleva al usuario de vuelta a la página de dónde vino?	0		
35	¿El sitio no deshabilita el botón "Atrás" y dicho botón aparece activo en la barra de herramientas del navegador en todas las páginas?	0		
36	¿El sitio evita que los usuarios se registren de manera innecesaria?	0		
37	¿La página se ve con cualquier resolución de pantalla?		x	
38	¿Es posible aumentar y disminuir el tamaño de letra?		x	

39	¿Las imágenes de los productos se pueden ampliar?		x	
40	¿Ofrece el contenido en otros formatos, como dispositivos móviles?	0		
41	¿Es posible imprimir la web sin perder información?	0		
42	¿Se pueden guardar las páginas web?		x	
43	¿El sitio provee una clara retroalimentación cuando una tarea ha sido completada exitosamente?	0		
44	¿Una imagen que sirve como enlace es fácilmente distinguible?		x	
45	¿En caso que se muestre información relacionada con registros obtenidos de la base de datos existe un sistema de navegación donde el usuario pueda especificar cuantos elementos desea ver en la página?		x	
46	¿Los usuarios son informados si es necesario un plug-in del navegador o resolución específico?		x	
47	¿Las páginas que utilizan nuevas tecnologías siguen funcionando cuando dicha tecnología no está presente (por ejemplo, los plug-ins de Flash)?		x	

48	¿Cuándo es necesaria la descarga de un plug-in, hay un enlace a la página donde obtenerlo?		x	
49	¿El motor de búsqueda maneja correctamente (No arroja ningún resultado) las búsquedas vacías (cuando no se introduce nada)?		x	
50	¿Las etiquetas de navegación y links contienen las “palabras clave” que los usuarios necesitan para alcanzar su objetivo?		x	
51	¿Los usuarios pueden ordenar y filtrar los resultados?	0		
52	¿Los enlaces que invocan acciones (ej. descargas, nuevas ventanas) están claramente distinguidos de los links que cargan otras páginas?	0		
53	¿La página de resultados de una búsqueda indica claramente cuántos resultados tuvo la búsqueda?	0		
54	¿La página de resultados de una búsqueda no muestra resultados duplicados (ni duplicados reales ni duplicados muy parecidos)?		x	
55	¿La caja de búsqueda es suficientemente grande para manejar la longitud		x	

	de las consultas más comunes?			
56	¿Las búsquedas cubren todo el sitio, no una porción de él?		x	
57	¿La interfaz de búsqueda está ubicada donde los usuarios esperan encontrarla (en la parte superior derecha de la página)?	0		
Consistencia y estándares				
58	¿Las imágenes tienen tamaños adecuados que no dificultan el acceso a las páginas?		x	
59	¿Existe un cambio visible cuando el ratón apunta a algo clickeable (excluyendo los cambios de cursor)?	0		
60	¿Todos los botones "clickeables" son efectivamente presionables?	0		
61	¿Los íconos son visualmente y conceptualmente distintos pero mantienen una armonía?	0		
62	¿Para tareas similares, los diálogos, formularios son similares?	0		
63	¿Existe una clara distinción entre campos "requeridos" y "opcionales" en los formularios?	0		
64	¿Las preguntas en los formularios están agrupadas de manera	1		

	lógica y cada grupo tiene un título descriptivo?			
65	¿Los campos en los formularios contienen ayudas, ejemplos o modelos de respuestas para demostrar el dato que se debe introducir?	1		
66	¿El nombre de los botones de un formulario es adecuado, aplicado a la acción, no general (Ej.: utilizar "Enviar" en vez de "OK")?	0		
67	¿Las listas de opciones, botones de radio y casillas son preferibles a las cajas de texto en los formularios?	0		
68	¿Se mantiene una navegación consistente y coherente en todas las pantallas?	0		
69	¿La distribución y ubicación de los elementos estructurales que contienen las páginas se mantiene constante a lo largo de la aplicación?		x	
70	¿Se usa la misma fuente para todos los navegadores?		x	
71	¿Puede utilizarse en cualquier navegador?	0		
72	¿Se mantiene una tipografía coherente en todo el sitio web?	0		
73	¿Existen enlaces redundantes? Enlaces redundantes: Son		x	

	enlaces con rótulos diferentes que llevan a una misma página.			
74	¿El link al mapa del sitio aparece en todas las páginas del sitio?		x	
75	¿Añade una descripción en las imágenes?		x	
76	¿Señala claramente los vínculos a archivos PDF como tal?		x	
77	¿Especifica el tamaño de los archivos PDF?		x	
78	¿El sitio tiene una URL correcta, clara y fácil de recordar?	0		
79	¿Tiene un tiempo de respuesta rápida?	0		
80	¿Los vínculos llevan a donde prometen ir?	0		
81	¿El sistema de navegación es amplio y sencillo (muchos ítems en un menú) en vez de un menú profundo?	0		
82	¿Se usan nombres estandarizados (“mapa web”, “acerca de...”)?	0		
83	¿Se proporciona un texto equivalente para todo elemento no textual, tales como imágenes, para explicar su contenido a discapacitados visuales?		x	
84	¿El documento está estructurado para que pueda ser leído con o sin una hoja de estilo, utilizando		x	

	adecuadamente los tags de HTML?			
85	¿Las presentaciones multimedia, en caso de existir están sincronizadas con sus subtítulos?		x	
86	¿El cambio de idioma en los textos está identificado? El texto que se encuentre en otro idioma debe estar en cursiva o en un formato diferente al del resto del texto.		x	
87	¿El sitio funciona correctamente en un lector de pantalla y/o navegador de voz?		x	
88	¿Es posible la navegación sin ratón?		x	
89	¿Se han creado atajos de teclado?		x	
90	¿Todos los estilos se han creado en hojas CSS?	0		
Estética y diseño minimalista				
91	¿Cumple el sitio con el principio de usabilidad de realizar las operaciones con un máximo de tres click?	0		
92	¿Existe suficiente contraste entre el color del fondo y el del texto?	0		
93	¿Los tipos y tamaños de letra son legibles y distinguibles?	0		
94	¿Añade color de fondo a los div que llevan imagen		x	

	de fondo? (Para los usuarios que desactivan las imágenes, desaparece el contraste entre texto y fondo, convirtiéndose en texto ilegible.)			
95	¿Poseen las páginas animaciones innecesarias?		x	
96	¿El sitio puede ser usado sin desplazamiento horizontal?	0		
97	¿El uso de los colores es moderado?	0		
98	¿Se usan los estilos (negritas, cursivas...) con moderación? Si todo está resaltado con negrita o cursiva, el cerebro se acostumbra y deja de parecerle destacado.	0		
99	¿Utiliza la misma tipografía los mismos aspectos, ejemplo: mantener una tipografía estándar por elementos?	0		
100	¿Utiliza un interlineado adecuado para una buena lectura?	0		
101	¿Se usan frases breves y concisas: que resuman los puntos clave y vayan al grano?		x	
102	¿Los párrafos son cortos?		x	
103	¿Los textos están corregidos?		x	

104	¿Resalta en negrita los conceptos principales de los textos densos?		x	
105	¿Utiliza listas de boliche y numeradas?	0		
106	¿El sitio evita el uso excesivo del texto en mayúsculas?	0		
107	¿El texto dentro del sitio no debe estar justificado?	0		
108	¿No se utiliza el texto subrayado en el cuerpo del texto para resaltar a menos que sea un hipervínculo?	0		
Prevención de errores				
109	¿Existe suficiente espacio entre los elementos de acción (links, botones, etc.) para prevenir que el usuario haga click en el elemento incorrecto?	0		
110	¿Hay ausencia de enlaces rotos o que no lleven a ninguna página?	0		
111	¿Se dan indicaciones para completar campos problemáticos?	0		
112	¿Los botones de acción, (tales como "Enviar") siempre son invocados por el usuario y no automáticamente invocados por el sistema cuando el último campo de un formulario ha sido lleno?	0		
113	¿El espacio entre los campos del formulario es	0		

	suficiente como para distinguirlos unos de otros?			
114	¿Se evita el contenido importante del sitio en ventanas emergentes?	0		
115	¿La zona de acción del vínculo esta ampliada para acertar a la primera?	0		
116	¿El buscador (si existe) permite errores tipográficos y ortográficos (tildes)?		x	
Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores				
117	¿En caso de errores de consistencia dentro del sitio, ¿se ofrece un mensaje de personalizado mediante una página explicativa?, (Por ejemplo: Error 404 para página inexistente)		x	
118	¿El mensaje de error permite volver a la situación anterior?	0		
119	¿Entrega información de contacto con múltiples opciones dentro y fuera de Internet? (Por ejemplo: Formulario, Teléfono, Fax, Correo electrónico, Chat, y hasta Wave, Buzz, Skype..., teléfono institucional, mesa de ayuda)		x	
120	¿Ofrece área de Preguntas Frecuentes		x	

	con datos de ayuda a usuarios?			
121	¿Ofrece páginas de ayuda que explican cómo usar el Sitio?		x	
122	¿Si la ayuda obliga a salir de la zona principal, se proporciona un medio para moverse entre esa ventana y la ayuda?		x	
123	¿La ayuda no interrumpe la tarea del usuario?		x	
124	¿El sitio está diseñado para necesitar el mínimo de ayuda y de instrucciones?	0		
125	¿La ayuda está organizada en pasos?	0		
126	¿Se dan ejemplos para facilitar la tarea?	0		
127	¿Se utilizan explicaciones cortas en la ayuda?	0		
Ayuda y documentación				
128	¿Existe un vínculo a los datos de contacto en un lugar bien visible en todas las páginas web del sitio?		x	
129	¿Información necesaria en página de contacto? (Atención al cliente, Consultas, Soporte técnico, Solicitudes de empleo.)		x	
130	¿Existe una página con las indicaciones sobre la protección de datos en el sitio web?		x	

131	¿Incluye un vínculo a los datos legales en todas las páginas?		x	
132	¿Puede el usuario ponerse en contacto con el encargado del Sitio Web para hacer sugerencias o comentarios?		x	
133	¿Funcionan correctamente los formularios de contacto?		x	
134	¿Hay alguien encargado de recibir y contestar estos mensajes?		x	
135	¿El sitio soporta a los usuarios novatos y expertos brindando diferentes niveles de explicación? (ej. en páginas de ayuda y mensajes de error)	Si		
136	¿La política de privacidad del sitio es fácil de encontrar, especialmente esas páginas que piden información personal?		x	
137	¿Cuándo existen múltiples pasos en una tarea, el sitio muestra todos los pasos que deben ser completados y provee una retroalimentación al usuario indicándole la posición actual en toda la ruta de la tarea?	Si		
138	¿La funcionalidad de los controles para nuevos dispositivos es exactamente la misma		x	

	que para los otros dispositivos?			
Flexibilidad y eficiencia de uso				
139	Se defina de manera correcta gráficos y tablas utilizando atributos (leyendas, unidades de medida, etc.)		x	
140	¿Las partes o secciones más importantes de los sitios son accesibles desde la página de inicio?	0		
141	¿Las páginas no requieren volver a escribir la información solicitada en páginas anteriores?	0		
142	¿Existen aceleradores, accesos rápidos a operaciones frecuentes?	0		
143	¿El cursor se desplaza adecuadamente en un formulario al presionar "tabulador"?	0		
144	¿Se implementen validaciones antes de que el usuario envíe información?	0		