



**Universidad de Matanzas**  
**Facultad de Ciencias Técnicas**  
**Especialidad Ingeniería Informática**

**Trabajo de Diploma en opción al título de Ingeniero Informático**

**Título:** “Aplicación web de apoyo a la gestión de la información del Departamento de Preparación y Superación de Cuadros de la Universidad de Matanzas”.

**Autora:** Zuliam Cruz Rodríguez.

**Tutores:** Ms.C. Julio L. Betancourt Ávila

Dr.C. Mailé Salgado Cruz

**Matanzas, 2018**

## **Declaración de autoría**

Declaro ser el único autor de este trabajo y autorizo al Departamento de Informática de la Universidad de Matanzas a hacer uso del mismo.

Para que así conste firmo la presente a los 30 días del mes de mayo de 2018.

Firma del Autor

---

Zuliam Cruz Rodríguez

## ***Pensamiento***

*"El futuro de nuestro país tiene que ser necesariamente un futuro de hombres de Ciencia, un futuro de hombres de pensamiento."*

*Fidel Castro Ruz*

*A todo puedo hacerle frente, gracias a Cristo que me fortalece.*

*Filipenses 4:13.*

*Cuando quieres algo, todo el universo conspira para que realices tu deseo.*

*Paulo Coelho*

## *Dedicatoria*

*Dedico esta investigación a todos aquellos que me han apoyado de manera incondicional a la realización de este trabajo y el haber depositado su confianza en mí.*

*A mis Padres y Esposo.*

## *Agradecimientos*

*A Dios, porque mi fe me ha llevado a donde estoy hoy.*

*A mis padres a los cuales les debo todo lo que soy, por su apoyo y confianza, siempre les estaré eternamente agradecida.*

*A mi esposo por su apoyo incondicional en estos años de grandes esfuerzos y que tuvo que escucharme hablar durante meses de la tesis.*

*A mi familia, mis hermanos de la fe, mis abuelas, mis tías y tíos, a mis primos y primas y a todos quienes de una forma u otra hicieron posible este sueño en mi vida.*

*A mis tutores Julio y Mailé por su paciencia, ayuda y dedicación en todo momento.*

*A todos los profesores que han contribuido a mi formación en el transcurso de mi carrera.*

*A las amistades que me ayudaron en todo momento Lianet, Heidy, Lorena, Ediel, Roberto, Geikel y los demás del grupo.*

*A todo aquel que hizo posible que pudiera graduarme.*

## **Resumen**

El Departamento de Preparación y Superación de Cuadros de la Universidad de Matanzas fue creado con el propósito de asesorar la preparación y supervisión de cuadros en el territorio matancero, así como la dirección por objetivos, la planeación estratégica de la universidad, el desarrollo local, e impartir en pregrado y posgrado las temáticas de dirección con un claustro revolucionario de reconocido nivel científico y prestigio nacional e internacional, que contribuya al desarrollo sostenible del territorio matancero. En la actualidad estos procesos se realizan de forma manual y no están informatizados, lo que dificulta mantener los mismos de una manera eficiente y asequible a todo el personal del departamento, de la universidad y del territorio. La aplicación web servirá para lograr una mejor visualización y manejo de la información del departamento que posibilita el conocimiento del mismo en la universidad y en el territorio. En una primera etapa se gestionará parte de la información de algunos de sus procesos como claustro, eventos y posgrado. La información debe ser accesible a las entidades del territorio, vinculado con la preparación de cuadros y reservas y el desarrollo local en Matanzas. En este trabajo se empleó tecnología de desarrollo de aplicaciones web con bases de datos y se desarrolló con enfoque de metodología de investigación.

## **Summary**

The Department of Preparation and Overcoming of Tables of the University Matanzas was created with the purpose and supervision of cadres in the Matanzas territory, as well as the strategic planning of the university, the local development and to teach in undergraduate and postgraduate the thematic of direction with a revolutionary cloister of level and national and international prestige, that contributes to the sustainable development of the territory of Matanzas. Currently these processes are carried out manually and are not computerized, which makes it difficult to maintain them in an efficient and affordable way to all the staff of the department, the university and the territory. The web application will serve to achieve a better visualization and management of the information of the department that allows the knowledge of the same in the university and the territory. In a first stage part of the information will be managed like cloister, events and postgraduate. The information must be accessible to the preparation of tables and reserves and local development in Matanzas. In this work, web application development technology was used with databases and was developed with a research methodology approach.

## Índice General

|  |    |
|--|----|
| Introducción.....  | 13 |
| Capítulo I: Fundamentación teórica y tendencias tecnológicas.....                | 17 |
| I.1.- Introducción. ....   | 17 |
| I.2.- Objeto de estudio.....   | 17 |
| I.2.1.- Objetivos estratégicos de la organización y misión. ....                 | 17 |
| I.2.2.- Descripción de las definiciones y procesos fundamentales a emplear. 18   |    |
| I.2.3.- Flujo actual de los procesos involucrados en el campo de acción. ....    | 18 |
| I.2.4.- Análisis crítico de los procesos involucrados en el campo de acción. ... | 19 |
| I.3.- Fundamentación teórica. ....   | 19 |
| I.3.1.- Antecedentes del trabajo. ....   | 20 |
| I.4.- Fundamentación de la metodología utilizada.....                            | 20 |
| I.4.1 Programación extrema.....  | 23 |
| I.5.- Aplicaciones web. ....   | 27 |
| I.6.- Patrón arquitectónico: Modelo Vista Controlador.....                       | 27 |
| I.7.- Tecnologías, herramientas y lenguajes de desarrollo.....                   | 28 |
| I.7.1.- Framework PHP Symfony. ....  | 29 |
| I.7.2.- Lenguajes de programación. ....  | 30 |
| I.7.3.- Sistemas de Gestión de Base de Datos. ....                               | 33 |
| I.7.4.- Servidor web.....  | 34 |
| I.8.- Conclusiones parciales del capítulo I.....                                 | 35 |
| Capítulo II: Análisis, diseño y construcción de la solución propuesta.....       | 36 |
| II.1.- Introducción ....   | 36 |
| II.2.- Etapa de Planificación. ....  | 36 |
| II.3.- Equipo y roles de trabajo. ....   | 36 |

|   |    |
|---|----|
| II.4.- Historias de Usuario iniciales.....  | 37 |
| II.5.- Planificación de iteraciones y plan de entrega.....                                      | 42 |
| II.5.1.- Planificación de iteraciones.....  | 42 |
| II.5.2.- Plan de entregas.....  | 43 |
| II.6.- Etapa de Diseño.....   | 44 |
| II.6.1.- Prototipo de la interfaz.....  | 45 |
| II.6.2.- Metáfora.....  | 45 |
| II.6.3.- Tareas de ingeniería.....  | 45 |
| II.6.4.- Modelo Físico de la Base de Datos.....   | 51 |
| II.7.- Conclusiones parciales del capítulo II.....  | 51 |
| Capítulo III: Pruebas a la aplicación web de apoyo a la gestión de la información del DPSC..... | 52 |
| III.1 Introducción.....   | 52 |
| III.2 Pruebas al software.....  | 52 |
| III.3 Pruebas de Aceptación.....  | 53 |
| III.4 Análisis de los resultados obtenidos en las pruebas.....                                  | 58 |
| III.5 Estimación del software.....  | 58 |
| III.6 Beneficio tangible e intangible.....  | 59 |
| III.7 Conclusiones parciales del capítulo.....  | 59 |
| Conclusiones.....   | 60 |
| Recomendaciones.....  | 61 |
| Referencias bibliográficas.....   | 62 |
| Anexos.....   | 65 |
| Anexo No. 1: Portada del Dossier del DPSC de la Universidad de Matanzas.....                    | 65 |
| Anexo No. 2: Convocatoria de evento del DPSC de la Universidad de Matanzas.....                 | 66 |

|   |    |
|---|----|
| Anexo No. 3: Interfaz de Gestionar usuario..... | 67 |
| Anexo No. 4: Interfaz de rol Departamento.....  | 67 |

## Índice de tablas

|   |    |
|---|----|
| Tabla 1: Roles del equipo de trabajo.....                 | 37 |
| Tabla 2: Historias de usuario iniciales.....              | 38 |
| Tabla 3: HU 1: Diseño y creación de la Base de Datos..... | 39 |
| Tabla 4: HU 2: Diseño de la interfaz de usuario.....      | 40 |
| Tabla 5: HU 3: Autenticarse.....                          | 40 |
| Tabla 6: HU 5: Gestionar Profesor.....                    | 41 |
| Tabla 7: HU 9: Gestionar Posgrado.....                    | 41 |
| Tabla 8: HU 9: Gestionar Evento.....                      | 42 |
| Tabla 9: HU 13: Gestionar Asesoría.....                   | 42 |
| Tabla 10: Tareas de Ingeniería.....                       | 46 |
| Tabla 11: TI 1: Diseño de la Base de Datos.....           | 47 |
| Tabla 12: TI 2: Creación de la Base de Datos.....         | 48 |
| Tabla 13: TI 5: Autenticarse.....                         | 48 |
| Tabla 14: TI 6: Insertar Usuario.....                     | 48 |
| Tabla 15: TI 9: Insertar profesor.....                    | 48 |
| Tabla 16: TI 22: Insertar posgrado.....                   | 49 |
| Tabla 17: TI 31: Insertar Evento.....                     | 49 |
| Tabla 18: TI 35: Imprimir Evento.....                     | 49 |
| Tabla 19: TI 37: Editar Asesoría.....                     | 49 |
| Tabla 20: TI 43: Eliminar Tareas.....                     | 50 |
| Tabla 21 TI 46: Eliminar entidad.....                     | 50 |
| Tabla 22 TI 47:Insertar currículum.....                   | 50 |
| Tabla 23 TI 52 :Editar bibliografía.....                  | 50 |
| Tabla 24: PA 3: Autenticarse.....                         | 54 |
| Tabla 25: PA 4: Gestionar usuario.....                    | 54 |
| Tabla 26 PA 5: Gestionar profesor.....                    | 55 |
| Tabla 27: PA 6: Gestionar clasificación.....              | 55 |
| Tabla 28: PA 7 Gestionar posgrado.....                    | 56 |
| Tabla 29: PA 10 Gestionar nota.....                       | 56 |
| Tabla 30: PA 11 Gestionar evento.....                     | 57 |
| Tabla 31 T: PA 12 Gestionar asesoria.....                 | 57 |

## Índice de Ilustraciones

|  |    |
|--|----|
| Ilustración 1: Procesos del DPSC de la Universidad de Matanzas ..... | 19 |
| Ilustración 2 Planificación de iteraciones por semana .....          | 43 |
| Ilustración 3 Plan de entregas .....                                 | 44 |
| Ilustración 4 Interfaz de usuarios .....                             | 45 |

## Introducción

El Departamento de Preparación y Superación de Cuadros (DPSC), pertenece a la Universidad de Matanzas, por lo cual es un departamento que realiza trabajo de docencia y de investigación, así como extensionista, que satisface necesidades de conocimientos, el desarrollo de habilidades y la solución de problemas en la esfera de la dirección y la gestión. (Salgado, 2018)

Para satisfacer los requerimientos de las entidades que acuden en busca de colaboración, asume como pauta de trabajo, sustentar su respuesta en diseños de “Programas a la Medida” adaptados a las condiciones y exigencias de nuestros clientes. En función de ello, el centro está en condiciones de acordar diferentes modalidades y tipos de programas de capacitación en las diversas temáticas de la Administración, en dependencia de las necesidades específicas de las instituciones solicitantes. (Salgado, 2018)

En la actualidad el DPSC gestiona la información y el intercambio con las estructuras de la provincia implicados con los procesos de preparación y supervisión de cuadros y reservas, como la planeación estratégica y el desarrollo local, de forma manual, por medio del correo electrónico y empleando herramientas ofimáticas, que dificultan su trabajo en la práctica diaria, demandando de más recursos para su realización tanto humanos como materiales.

En este sentido, se hace necesario buscar alternativas que contribuyan a resolver los problemas presentados en la gestión de la información del DPSC, en función de los procesos que ejecuta.

De lo anteriormente referido se identificó como **problema científico** el siguiente: ¿Cómo facilitar la gestión de la información del Departamento de Preparación y Superación de Cuadros de la Universidad de Matanzas?

Como ya es habitual, las entidades incorporan las Tecnologías de la Información y las Comunicaciones (TIC), como herramienta de apoyo en la gestión de la información de los procesos que ejecutan (Betancourt y García, 2016).

Una alternativa de solución a este problema sería una aplicación web de apoyo a la gestión y control de información del DPSC de la Universidad de Matanzas, en que se organizaría mejor la información, liberando espacio para trabajar, permitiría un acceso más rápido y fácil, una seguridad más eficaz y la transferencia de la información sería mucho más fácil y cómoda, el intercambio entre los diferentes actores de la provincia vinculado a la preparación y superación de cuadros y reservas, así como la visualización de las opciones para dar respuesta a los sistemas diseñados para la organización y entidades y los proyectos de desarrollo local y conocer los resultados que se obtiene en estas esferas de trabajo.

Como supuesto de solución al problema se plantea la siguiente **hipótesis**: si se desarrollara una aplicación web para el DPSC, entonces se facilitaría la gestión de la información de los procesos que ejecuta.

Para llevar a cabo este trabajo se propuso los siguientes objetivos.

El **objetivo general** es: desarrollar una aplicación web de apoyo a la gestión de la información del DPSC de la Universidad de Matanzas.

Los **objetivos específicos** son los siguientes:

1. Realizar un estudio de los fundamentos teóricos-metodológicos de los procesos que ejecuta el DPSC.
2. Diseñar una aplicación web para la gestión de información del DPSC.
3. Implementar una aplicación web para la gestión de información del DPSC.
4. Validar la aplicación web para la gestión de información del DPSC.

Para la realización de este trabajo se emplearon diferentes **métodos** tanto teóricos como empíricos, que han sido de gran ayuda en la investigación para el desarrollo del proceso. (Sampieri, 2005)

Los métodos teóricos fundamentales utilizados fueron: el histórico-Lógico, el inductivo-deductivo y el de análisis-síntesis.

La observación y análisis de los antecedentes del problema permitió realizar un estudio sobre las diferentes propuestas de análisis, diseño e implementación de

sistemas relacionados con el proceso de departamentos de cuadros de otras entidades, que han sido desarrollados hasta la actualidad.

Inductivo-Deductivo: Este método fue utilizado pues a través del modo de razonamiento, que va de lo particular a lo general se logró profundizar y valorar hasta donde podía ser posible la construcción de la aplicación, cuáles eran las salidas de las informaciones, y cómo se respondía al cumplimiento de los objetivos de la investigación.

Análisis-Síntesis: Se emplea este método pues a través de él se llega a un consenso de lo que se mostrará en la aplicación, también se tuvo en cuenta para el diseño y la implementación de los modelos y metodologías utilizadas.

Los métodos empíricos fueron: la observación científica, la entrevista, el análisis de documentos

La observación científica ha acompañado la investigación desde sus primeros momentos, principalmente en su desarrollo e implementación. Permitió investigar los procesos externamente sin tener que llegar a la esencia de los mismos, lo cual ayudó a realizar un seguimiento del comportamiento de estos procesos sirviendo como guía. Se aplicó sobre todo basado en el flujo actual de los procesos llevados a cabo por el departamento.

Las entrevistas se realizaron al personal del DPSC, vinculado con el desarrollo de este trabajo.

El análisis de documentos se empleó en la revisión presentada de documentos importantes del departamento, además de procedimientos de diferentes autores de cómo se desarrolla en la actualidad en proceso de posgrado. Se utilizó, además, para la fundamentación teórica y metodológica de la investigación.

El Trabajo de diploma se estructuró de la siguiente manera:

**Capítulo I:** “Fundamentación teórica y tendencias tecnológicas”. En este capítulo se refleja la base teórica de la investigación, donde se recoge los conceptos fundamentales y los elementos teóricos asociados con el problema a solucionar. El mismo describe los antecedentes del trabajo y cuál es su objeto de estudio.

También se exponen las tendencias tecnológicas actuales a utilizar en el mismo y la metodología de desarrollo de software.

**Capítulo II:** “Descripción de la propuesta”. En este capítulo se aborda la metodología utilizada en el trabajo, así como una propuesta de solución para la aplicación web y se presenta el proceso de planificación, teniendo en cuenta el equipo de trabajo, los roles, las historias de usuario, el plan de iteraciones y el plan de entrega.

**Capítulo III:** “Validación de la Solución Propuesta”. En este capítulo se incluye la descripción de las iteraciones, las tareas desplegadas y las pruebas aceptación. También, se presenta el estudio de los costos y beneficios.

## Capítulo I: Fundamentación teórica y tendencias tecnológicas.

### Capítulo I: Fundamentación teórica y tendencias tecnológicas.

#### I.1.- Introducción.

En este capítulo se presentan aspectos de carácter teórico sobre conceptos relacionados con el tema de investigación y se explica el flujo de los procesos que intervienen en la dinámica del Departamento de Preparación y Superación de Cuadros (DPSC). Por otra parte, se realiza un estudio de las tecnologías más usadas y sus tendencias actuales para el diseño e implementación de aplicaciones web, mediante una metodología de desarrollo para la modelación y la descripción de la propuesta y el lenguaje de programación, fundamentándose así la implementación de esta aplicación web.

#### I.2.- Objeto de estudio.

El objeto de estudio de esta investigación es la gestión de la información del DPSC, que pertenece a la Universidad de Matanzas, por tanto, surge la necesidad de hacer una descripción más detallada de estos elementos que se pretenden transformar y perfeccionar.

El campo de acción es, entonces, la informatización de algunos procesos que ejecuta el DPSC de la Universidad de Matanzas.

##### I.2.1.- Objetivos estratégicos de la organización y misión.

Entre los objetivos estratégicos del DPSC se encuentran asesorar la dirección por objetivos y la planeación estratégica de la universidad, así como la superación de cuadros y reservas del territorio.

La misión del DPSC está dirigida a formar cuadros y reservas; así como profesionales en pregrado y posgrado, en las temáticas de dirección; con un claustro revolucionario de reconocido nivel científico y prestigio nacional e internacional, que contribuya al desarrollo sostenible del territorio matancero.

I.2.2.- Descripción de las definiciones y procesos fundamentales a emplear.

Para una mejor comprensión del problema es necesario dominar los términos de posgrado, cuadro, asesoría, planeación estratégica, cronología, etc.

**Posgrado:** Es el ciclo de estudio de especialización que se cursa tras la graduación o licenciatura. Se trata de la formación que sigue el título de grado, que es titulación de educación superior que se obtiene al concluir una carrera universitaria de cuatro años. ([https://definicion.de/posgrado /](https://definicion.de/posgrado/)) (ver la portada del Dossier en el anexo 1)

**Cuadro:** Es el trabajador que posee la capacidad técnico-laboral requerida, una sólida preparación y habilidad dirección, disciplina administrativa, constante exigencia, vela por el uso correcto de los recursos puestos a su disposición, cuenta con un conjunto de cualidades político-ideológicas y éticas, asume los principios consagrados en nuestra constitución y el programa del Partido es elegido o designado para ocupar cargos de dirección u otro que por la importancia de las funciones que cumple lo requieran, en los Órganos Superiores, demás órganos Estatales, Entidades Presupuestadas y el Sistema Empresaria: Folleto “Código de ética de Cuadros”.

**Asesoría:** Es un servicio que consiste en brindar información a una persona real o jurídica. Mediante la misma se busca dar un respaldo en un tema que se conoce con gran detalle. (<https://definicion.me/asesoria/>)

**Planeación estratégica:** Es la elaboración, desarrollo, y puesta en marcha de distintos planes operativos por partes de las empresas u organizaciones con la intención de alcanzar objetivos y metas planteadas. Estos planes pueden ser a corto, mediano y largo plazo. ([https://definicion.de/ planeación estratégica /](https://definicion.de/planeacion-estrategica/))

I.2.3.- Flujo actual de los procesos involucrados en el campo de acción.

Los procesos involucrados en el campo de acción (ver la ilustración 1) son la gestión de la información del claustro del departamento, del trabajo con asignaturas en pregrado, el proceso de solicitud de los posgrados que incluye el intercambio entre las entidades y el departamento, la promoción de los eventos

(ver ejemplo de convocatoria en la ilustración 2 en el anexo 2), la planeación estratégica y los proyectos de investigación.



Ilustración 1: Procesos del DPSC de la Universidad de Matanzas.

El departamento de Preparación y Superación de cuadros de la universidad de Matanzas fue creado con el propósito de asesorar la preparación y supervisión de cuadros en el territorio matancero así como la dirección por objetivos, la planeación estratégica de la universidad, el desarrollo local, e impartir en pregrado y posgrado las temáticas de dirección, que contribuya al desarrollo sostenible del territorio matancero.

#### I.2.4.- Análisis crítico de los procesos involucrados en el campo de acción.

Atendiendo a lo anteriormente expuesto, se puede afirmar que gran parte de la información con que se trabaja en el DPSC está en hojas de papel, mientras que otra parte está en formato digital de herramientas ofimáticas. En la promoción de los eventos se dificulta en dar a conocer los sucesos que ofrece el comité organizador de dicha actividad, o al acceder a la información de estos para poder participar. También se dificulta la participación de las entidades en la solicitud de posgrados, tan necesarios para una buena dirección de las entidades.

#### I.3.- Fundamentación teórica.

En la sociedad actual, con el adelanto de la ciencia y la tecnología, las empresas en su mayoría se han visto envueltas en la informatización de sus procesos. Esto

permite una mejor gestión de la información incurriendo en un mejor control, seguridad y agilizando los procesos de manera considerable. El DPSC no se encuentra alejado de estos procesos, considerando de suma importancia la necesidad de una aplicación web que contribuya a la gestión de la información, además de brindar promoción para los eventos que realiza dicho departamento.

### I.3.1.- Antecedentes del trabajo.

Como antecedentes se ha tenido en cuenta que existen investigaciones vinculadas con el objeto de estudio, las que con diferentes enfoques abordan aplicaciones web de departamentos que preparan cuadros. Entre ellas se encuentran:

- Aplicación web de Dirección de Cuadros /Universidad de Oriente. Fue puesta en marcha el 12 de septiembre del 2016.
- Tesis de Grado en la Universidad Central de “Marta Abreu, que fue solo con alcance al Departamento de Ingeniería Industrial para evaluar íntegramente esta actividad, y que no se estableció ningún software.

Estos dos resultados no satisfacen los requerimientos del DPSC, en la Universidad de Matanzas, debido a que el alcance es aún mayor y tienen implementados los flujos de los procesos que se ejecutan.

### I.4.- Fundamentación de la metodología utilizada.

El auge de la tecnología ha originado la necesidad de implantar metodología de desarrollo de software, que tiene el objetivo de agilizar y automatizar los procesos. Estas ayudan a entregar un producto de calidad en tiempo y costo estimados (Anexo 1). Las metodologías ágiles de desarrollo de software han despertado interés gracias a que proponen simplicidad y velocidad para crear sistemas (Highsmith, 2001).

Las metodologías tradicionales no se adaptan a las nuevas necesidades o expectativas que tienen los usuarios, en parte porque los métodos usados no son flexibles ante la posibilidad de la exigencia de nuevos requerimientos (Cao &

otros., 2003). Aunque esto está en dependencia de las dimensiones del problema del negocio a resolver.

Los cambios generalmente implican altos costos, demanda de tiempo y la reestructuración total del proyecto que se esté llevando; en contraparte, los métodos ágiles permiten un desarrollo iterativo y adaptable que permite la integración de nuevas funcionalidades a lo largo del desarrollo del proyecto; para que tanto el cliente como el desarrollador queden satisfechos porque el producto final tiene una calidad adecuada (Giraldo, 2006).

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

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

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

El manifiesto ágil plantea que es necesario valorar más a los individuos y su interacción que a los procesos y las herramientas. Las herramientas mejoran la eficiencia, pero sin personas con conocimiento técnico y actitud adecuada, no producen resultado (Letelier, Patricio, Penadés, María Carmen y Canós, & H., 2006).

También afirma que en la mayoría de las ocasiones se crea el entorno y luego se espera a que el equipo se adapte a él, lo cual constituye un error que debe ser

erradicado, ya que debe ser el equipo y no de forma contraria, el que en base a sus necesidades sea capaz de crear el entorno. Destaca la importancia de poder ver anticipadamente cómo se comportan las funcionalidades esperadas sobre prototipos o sobre las partes ya elaboradas del sistema final. El manifiesto ágil define al cliente como un miembro más del equipo, que se integra y colabora en el grupo de trabajo (José H, Canós, Letelier, Patricio y Penadés, & Carmen., 2003).

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

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

- En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

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

En este trabajo se empleó la metodología ágil de Programación Extrema.

#### I.4.1 Programación extrema.

Programación extrema (*eXtreme Programming*, XP por sus siglas en inglés) 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 (Joskowicz, 2008). XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios (Goto, Tsuchida, Kensei y Nishino, & Tetsuro., 2014). XP se define como especialmente adecuada para proyectos con requisitos imprecisos donde existe un alto riesgo técnico (Beck, 2000).

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

##### **Historias de Usuario.**

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

Es la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características

que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (Jeffries, Anderson, & Hendrickson, 2001).

### **Roles XP.**

Los roles de acuerdo con la propuesta original son:

- Programador: escribe las pruebas unitarias y produce el código del sistema. Responsable de decisiones técnicas, de construir el sistema, sin distinción entre analistas, diseñadores o codificadores (Hurtado, Ariel, & Bastiarrica, 2005).
- Cliente: escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio (Calero, 2003).
- Encargado de pruebas (*Tester*): ayuda al cliente a escribir las pruebas funcionales. (Reynoso, 2012). Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas (Beck, 2000).
- Encargado de seguimiento (*Tracker*): proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración (Cubel, Navarro, y Jose, & María, 2012).
- Entrenador (*Coach*): es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.
- Consultor: es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.

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

### **Proceso XP.**

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

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

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

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

### **Prácticas XP.**

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

- El juego de la planificación. Hay una comunicación frecuente el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo

requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.

- Entregas pequeñas. Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.
- Metáfora. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema, ayudando a la nomenclatura de clases y métodos del sistema).
- Diseño simple. Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.
- Pruebas. La producción de código está dirigida por las pruebas unitarias. Estas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema.
- Refactorización (*Refactoring*). Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.
- Programación en parejas. Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores).
- Propiedad colectiva del código. Cualquier programador puede cambiar parte del código en cuanto lo desee.
- Integración continua. Cada pieza de código es integrada en el sistema una vez que esté lista.
- 40 horas por semana. Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre,

probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo (Gittins, y, Hope, & Sian., 2001) .

- Cliente in-situ. El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Éste es uno de los principales factores de éxito del proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.
- Estándares de programación. XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible.

#### I.5.- Aplicaciones web.

En la ingeniería de software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los *webmails*, *wikis*, *weblogs*, tiendas en línea y la propia Wikipedia que son ejemplos bastante conocidos de aplicaciones web (Luján 2001).

En este trabajo se presenta una aplicación web para apoyar en la gestión de información de los procesos del DPSC.

#### I.6.- Patrón arquitectónico: Modelo Vista Controlador.

El modelo-vista-controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica del negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello, MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la

representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

El MVC es aplicable al desarrollo de cualquier aplicación independientemente del lenguaje de programación elegido. MVC consiste en dividir el código de una aplicación en capas (D. G. Fernández, 2012) :

**Modelo:** es todo acceso a datos y las funciones que llevan lo que llaman "lógica de negocio", o sea datos y reglas de negocio. Lleva un registro de las vistas y controladores del sistema. Cada acceso a datos se pone en su función individual porque, de esta forma, si se cambia de gestor de bases de datos este cambio sólo afecta a estas funciones, no al resto de la aplicación. Tener el modelo bien delimitado permite la existencia de varias aplicaciones que compartan el mismo.

**Vista:** en una aplicación web, es el HTML y lo necesario para convertir datos en HTML. O sea, muestra la información del modelo al usuario. Tienen un registro de su controlador asociado (normalmente porque además lo instancia). Pueden dar el servicio de "Actualización", para que sea invocado por el controlador o por el modelo. Tener la vista separada del controlador permite cambiar la aplicación para que genere, en lugar de HTML, algo distinto (por ejemplo, WML), sin tener que tocar más que una parte completamente delimitada del código.

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

#### I.7.- Tecnologías, herramientas y lenguajes de desarrollo.

Para el desarrollo de una solución eficiente es necesario realizar un estudio previo de las tecnologías existentes, a fin de seleccionar las más adecuadas para la tarea

en cuestión. Sobre esta base se seleccionaron las herramientas tecnológicas que se muestran.

#### I.7.1.- Framework PHP Symfony.

Un framework simplifica el desarrollo de las aplicaciones, ya que automatiza muchos de los patrones utilizados para resolver las tareas comunes. Además, proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. También, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web (*Symfony Documentation*, 2016).

Se decide seleccionar el Framework Symfony para desarrollar la propuesta de solución debido a que presentan características de mucha utilidad que facilitan en gran medida la realización de la aplicación. La estructura interna que posee Symfony permite separar cada componente que forma parte de la aplicación, lo que permite un fácil acceso a dichos componentes, tanto al desarrollador como a la persona encargada de ofrecer mantenimiento al software. También es muy favorable la facilidad con la que se puede desplegar la aplicación creada desde el entorno de desarrollo hacia el entorno de producción. Las características siguientes apoyan la decisión de la selección de Symfony como *framework* para el desarrollo de la propuesta de solución:

- Fácil de instalar y configurar en la mayoría de plataformas con la garantía de que funciona correctamente en los sistemas Windows y Unix.

## Capítulo I: Fundamentación teórica y tendencias tecnológicas.

- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- 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 que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros (Symfony Documentation, 2016).

### I.7.2.- Lenguajes de programación.

**PHP** (*Hypertext Pre-Processor*, Lenguaje de Programación Interpretado) es un lenguaje de código abierto muy popular creado por el programador informático Rasmus Lerdorf.

Es un lenguaje adecuado para desarrollo web y que puede ser incrustado en HTML (*HyperText Markup Language*, lenguaje de marcas de hipertexto). En lugar de usar muchos comandos para mostrar HTML, las páginas de PHP contienen HTML con código incrustado. ¿El código de PHP está encerrado entre las etiquetas especiales de comienzo y final `<? php y?>` que permiten entrar y salir del modo PHP. Lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script PHP, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga. Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores

profesionales. No sienta miedo de leer la larga lista de características de PHP. En unas pocas horas podrá empezar a escribir sus primeros scripts (Lerdorf, 2017).

PHP posee características que favorecen en gran medida el desarrollo de la aplicación propuesta, las cuales son:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad.
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- No requiere definición de tipos de variables, aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Permite el manejo de excepciones.

### **HTML (*Hypertext Markup Language*)**

Es un lenguaje sencillo con enlaces que conducen a otros documentos o fuentes de información relacionadas. La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.), así como los diferentes efectos que se quieren dar (cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado. Es uno de los lenguajes más utilizados a la hora de desarrollo web ya que responde a las

necesidades de manipulación y maquetación de los elementos de la aplicación (Álvarez R. , 2012).

Se decide utilizar este lenguaje ya que es un lenguaje que contiene gran variedad de etiquetas para la creación de páginas web que serían las vistas de la aplicación. Es compatible con la mayoría de los navegadores web y es compatible con el framework php utilizado para el desarrollo del trabajo.

### **JavaScript**

Es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con este lenguaje se pueden crear diferentes efectos e interactuar con usuarios o los diferentes componentes de una página web. Posee varias características, es basado en acciones que posee menos restricciones y también es multiplataforma. Gran parte de la programación está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros.

Es necesario resaltar que hay dos tipos de *JavaScript*: por un lado, está el que se ejecuta en el cliente, este es el *JavaScript* propiamente dicho, aunque técnicamente se denomina Navegador JavaScript. Pero también existe un JavaScript que se ejecuta en el servidor, es más reciente y se denomina *LiveWire JavaScript* (Valdés, 2007). En adición a lo que plantea Valdés, el funcionamiento de JavaScript se evidencia en gran medida en las acciones del usuario en una aplicación web ya que permite validar datos introducidos por él, mostrar mensajes relacionados con la acción solicitadas e incluso posibilita modificar la manera en que es presentada la información en las vistas.

### **CSS (*Cascading Stylesheets*, Hojas de Estilo en Cascada)**

Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y *XHTML*. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Al crear una página web, se utiliza en primer lugar el

lenguaje *HTML/XHTML* para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, entre otros. Una vez creados los contenidos, se utiliza el lenguaje CSS 3 para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página. Separa la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos *HTML/XHTML* bien definidos y con significado completo también llamados documentos semánticos. Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes (Eguiluz, 2017).

CSS modifica el aspecto visual del contenido haciéndolo más entendible o estándar. Adicionalmente también se pueden seguir pautas de diseño a la hora de modificar el estilo de las vistas al hacer uso de este lenguaje.

### I.7.3.- Sistemas de Gestión de Base de Datos.

Los Sistemas Gestores de Bases de Datos (SGBD, en inglés DBMS: *DataBase Management System*) son un tipo de software que permiten la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación (Marcano, 2009).

#### **MySQL**

Es un gestor de base de datos sencillo de usar e increíblemente rápido sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas también en esos lenguajes. Es uno de los sistemas gestores de bases de datos más utilizados en la actualidad, utilizado por grandes corporaciones como Yahoo! Finance, Google, Motorola, entre otras. Es gratis para aplicaciones no comerciales. Dentro de las principales características se encuentra, la gran

portabilidad entre sistemas, soporta gran cantidad de tipos de datos para las columnas y hasta 32 índices por tabla, manteniendo un buen nivel de seguridad pues gestiona los datos de los usuarios y contraseñas, dispone de API's en gran cantidad de lenguajes (Python, C, C++, Java, PHP) y aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multi-hilo (Martínez, 2014).

Se eligió MySQL, porque es uno de los servidores de bases de datos de código abierto más populares y conocidos del mundo, un sistema de manejo de bases de datos con un gran nivel de estabilidad y facilidad de desarrollo que se integra fácilmente con el lenguaje de programación PHP. Dispone, además, de una arquitectura que lo hace extremadamente rápido y fácil de personalizar. Sumándole a todos estos beneficios, que es un servidor que se adecúa perfectamente a las exigencias del cliente.

#### I.7.4.- Servidor web.

Un servidor web o servidor HTTP es un programa que, utilizando el modelo Cliente-Servidor y el protocolo HTTP, procesa una aplicación del lado del servidor, a través de conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente y genera o cede una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo (Falcón, 2006).

**XAMPP** es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Windows, como sistema operativo.
- Apache, como servidor web.
- MySQL, como gestor de bases de datos.
- PHP (generalmente), Perl, o Python, como lenguajes de programación (Quintero, 2012).

- PhpStorm es un potente entorno de desarrollo integrado (IDE), especialmente diseñado a fin de proporcionar a los desarrolladores de HTML, JavaScript y PHP todas las herramientas necesarias para su trabajo. PhpStorm proporciona un editor de código enriquecido e inteligente para PHP con resaltado de sintaxis, configuración extendida de formateo del código, navegación rápida y comprobación de errores sobre la marcha y finalización de código inteligente (Fernández, 2011).

#### I.8.- Conclusiones parciales del capítulo I.

En este capítulo se consolidaron las bases teóricas con vistas al desarrollo de la aplicación web, mediante un estudio de las principales herramientas y tecnologías que se proponen para el proyecto. Se profundizó en el análisis de las herramientas existentes hasta el momento, para concluir con los resultados que se esperan alcanzar al final del trabajo. Se definió el modelo de desarrollo a seguir, el cual describe cada artefacto a obtener en cada etapa del desarrollo de esta aplicación y se sientan las bases que dan pie a su inicio. Quedó plasmada de manera clara la necesidad de elaborar una herramienta informática de apoyo a la gestión de la información en el DPSC de la Universidad. Se justificó la utilización de una metodología ágil para el desarrollo de la aplicación, específicamente la metodología XP.

## Capítulo II: Análisis, diseño y construcción de la solución propuesta.

### Capítulo II: Análisis, diseño y construcción de la solución propuesta.

#### II.1.- Introducción

En el capítulo se exponen los elementos necesarios para la descripción de la solución propuesta para el DPSC. A través de las Historias de Usuarios (HU) que acumulan la necesidad existente definida por el cliente, se lleva a cabo el análisis de los requerimientos, donde se aplica la Metodología XP. Para ello se identifican varias etapas, que van desde la declaración del problema y los requerimientos del sistema, hasta las pruebas y la liberación del mismo. Las metodologías de desarrollo proveen de una guía que ayuda al grupo de desarrollo a organizarse en tiempo, actividades y artefactos a desarrollar.

#### II.2.- Etapa de Planificación.

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

#### II.3.- Equipo y roles de trabajo.

La metodología XP exige que el desarrollo se realice en parejas para agregar calidad al código que se escribe. Además, se tomará como principal fuente de aporte de ideas todas aquellas propuestas hechas por el cliente en cada una de las pruebas.

Se trabajan cuarenta horas semanales, plasmando el principio de que horas extras no aumentan la productividad, sino que desmotivan al equipo, programadores cansados son más propensos a cometer errores y son menos productivos.

## Capítulo II: Análisis, diseño y construcción de la solución propuesta.

A continuación, se presentan en la tabla 1 los roles del equipo de trabajo para el desarrollo del software, y los miembros involucrados en el desarrollo del mismo.

Tabla 1: Roles del equipo de trabajo.

| Miembros                  | Roles  |
|---------------------------|--|
| Zuliam Cruz Rodríguez     | Programador, Encargado de pruebas, Encargado de seguimiento. |
| Mailé Salgado Cruz        | Cliente, Encargado de pruebas.                               |
| Julio L. Betancourt Ávila | Entrenador, Gestor, Consultor.                               |

### II.4.- Historias de Usuario iniciales.

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

Son utilizadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si la aplicación cumple con lo que especifica la historia de usuario. Las HU ayudan en la comunicación entre el cliente y los desarrolladores y pueden ir cambiando a medida que avanza el proyecto y que el cliente vea nuevas posibilidades y soluciones. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia.

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

Escalas equivalentes a la prioridad en el negocio:

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

Capítulo II: Análisis, diseño y construcción de la solución propuesta.

**Media:** Dada a las HU 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 HU que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el proyecto en desarrollo.

El resumen, las HU iniciales que se muestra en la tabla 2, cubren los requerimientos que fueron planteados por el cliente, aunque la planificación es flexible ante los cambios que puedan ocurrir durante el desarrollo del proyecto.

Tabla 2: Historias de usuario iniciales.

| No | Historia de Usuario                    | Prioridad | Riesgo | Iteración | Puntos Estimados | Entrega |
|----|--|-----------|--------|-----------|------------------|---------|
| 1  | Diseño y creación de la base de datos. | Alta      | Alto   | 1         | 1                | 1       |
| 2  | Diseño de la interfaz de usuario.      | Alta      | Medio  | 1         | 1                | 1       |
| 3  | Autenticarse.                          | Alta      | Medio  | 1         | 1                | 1       |
| 4  | Gestionar usuarios                     | Alto      | Medio  | 1         | 3                | 1       |
| 5  | Gestionar profesores                   | Alta      | Medio  | 2         | 3                | 2       |
| 6  | Gestionar clasificaciones              | Alta      | Medio  | 2         | 2                | 2       |
| 7  | Gestionar tipos de posgrados.          | Alta      | Medio  | 2         | 1                | 2       |
| 8  | Gestionar posgrado.                    | Alta      | Medio  | 3         | 3                | 3       |
| 9  | Gestionar participante.                | Alta      | Medio  | 3         | 1                | 3       |
| 10 | Gestionar nota.                        | Alto      | Media  | 3         | 2                | 3       |
| 11 | Gestionar evento.                      | Alto      | Media  | 4         | 3                | 4       |

Capítulo II: Análisis, diseño y construcción de la solución propuesta.

|    |                        |      |       |   |   |   |
|----|------------------------|------|-------|---|---|---|
| 12 | Gestionar asesoría.    | Alto | Medio | 4 | 3 | 4 |
| 13 | Gestionar tareas.      | Alto | Medio | 4 | 2 | 4 |
| 14 | Gestionar entidad      | Alto | Medio | 5 | 1 | 4 |
| 15 | Gestionar currículo    | Alto | Medio | 5 | 1 | 5 |
| 16 | Gestionar bibliografía | Alto | Medio | 5 | 2 | 5 |
| 17 | Gestionar archivos     | Alto | Medio | 5 | 2 | 5 |
| 18 | Gestionar solicitudes  | Alto | Medio | 5 | 2 | 5 |

Se presenta una muestra de las HU de este trabajo, descrita a manera de ejemplo (ver tablas de la 3 a la 9).

Tabla 3: HU 1: Diseño y creación de la Base de Datos.

| <b>Historia de Usuario</b>  |                                   |
|---|-----------------------------------|
| <b>No:1</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre de la HU:</b> Diseño y creación de la Base de Datos.                  |                                   |
| <b>Prioridad en negocio:</b> Alta   | <b>Riesgo en desarrollo:</b> Alto |
| <b>Puntos estimados:</b> 1  | <b>Iteración asignada:</b> 1      |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.                          |                                   |
| <b>Descripción:</b> Se diseña e implementa la base de datos en el gestor MySQL. |                                   |
| <b>Observaciones:</b>   |                                   |

Capítulo II: Análisis, diseño y construcción de la solución propuesta.

Tabla 4: HU 2: Diseño de la interfaz de usuario.

| Historia de Usuario   |                                    |
|---|------------------------------------|
| <b>No:2</b>   | <b>Usuario:</b> Administrador      |
| <b>Nombre de la HU:</b> Diseño de la interfaz de usuario.               |                                    |
| <b>Prioridad en negocio:</b> Alta                                       | <b>Riesgo en desarrollo:</b> Medio |
| <b>Puntos estimados:</b> 1  | <b>Iteración asignada:</b> 1       |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.                  |                                    |
| <b>Descripción:</b> Se muestra las vistas de las diferentes interfaces. |                                    |
| <b>Observaciones:</b>   |                                    |

Tabla 5: HU 3: Autenticarse.

| Historia de Usuario   |                                    |
|---|------------------------------------|
| <b>No:3</b>   | <b>Usuario:</b> Administrador      |
| <b>Nombre de la HU:</b> Autenticarse.   |                                    |
| <b>Prioridad en negocio:</b> Alta   | <b>Riesgo en desarrollo:</b> Medio |
| <b>Puntos estimados:</b> 1  | <b>Iteración asignada:</b> 1       |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.  |                                    |
| <b>Descripción:</b> todos los usuarios se identifican por nombre y contraseña y se les otorgan los permisos según su rol. |                                    |
| <b>Observaciones:</b>   |                                    |

Capítulo II: Análisis, diseño y construcción de la solución propuesta.

Tabla 6: HU 5: Gestionar Profesor.

| Historia de Usuario  |                                    |
|--|------------------------------------|
| <b>No:5</b>  | <b>Usuario:</b> Administrador      |
| <b>Nombre de la HU:</b> Gestionar profesores.                                    |                                    |
| <b>Prioridad en negocio:</b> Alta  | <b>Riesgo en desarrollo:</b> Medio |
| <b>Puntos estimados:</b> 3   | <b>Iteración asignada:</b> 2       |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.                           |                                    |
| <b>Descripción:</b> se insertan, eliminan , muestran y modifican los profesores. |                                    |
| <b>Observaciones:</b>  |                                    |

Tabla 7: HU 9: Gestionar Posgrado.

| Historia de Usuario  |                                    |
|--|------------------------------------|
| <b>No:9</b>  | <b>Usuario:</b> Departamento       |
| <b>Nombre de la HU:</b> Gestionar posgrado.                                    |                                    |
| <b>Prioridad en negocio:</b> Alta  | <b>Riesgo en desarrollo:</b> Medio |
| <b>Puntos estimados:</b> 3   | <b>Iteración asignada:</b> 3       |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.                         |                                    |
| <b>Descripción:</b> se insertan, eliminan , muestran y modifican los posgrado. |                                    |
| <b>Observaciones:</b>  |                                    |

## Capítulo II: Análisis, diseño y construcción de la solución propuesta.

Tabla 8: HU 9: Gestionar Evento.

| Historia de Usuario   |                                    |
|---|------------------------------------|
| <b>No:12</b>  | <b>Usuario:</b> Departamento       |
| <b>Nombre de la HU:</b> Gestionar evento.                                     |                                    |
| <b>Prioridad en negocio:</b> Alta   | <b>Riesgo en desarrollo:</b> Medio |
| <b>Puntos estimados:</b> 3  | <b>Iteración asignada:</b> 4       |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.                        |                                    |
| <b>Descripción:</b> se insertan, eliminan , muestran y modifican los eventos. |                                    |
| <b>Observaciones:</b>   |                                    |

Tabla 9: HU 13: Gestionar Asesoría.

| Historia de Usuario  |                                    |
|--|------------------------------------|
| <b>No:13</b>   | <b>Usuario:</b> Departamento       |
| <b>Nombre de la HU:</b> Gestionar Asesoría.                                    |                                    |
| <b>Prioridad en negocio:</b> Alta  | <b>Riesgo en desarrollo:</b> Medio |
| <b>Puntos estimados:</b> 3   | <b>Iteración asignada:</b> 4       |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.                         |                                    |
| <b>Descripción:</b> se insertan, eliminan , muestran y modifican las asesoria. |                                    |
| <b>Observaciones:</b>  |                                    |

II.5.- Planificación de iteraciones y plan de entrega.

II.5.1.- Planificación de iteraciones.

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase y se genera al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración.

En la ilustración 3 se muestran las tres iteraciones y la cantidad de semanas en los que se demora el desarrollo de cada tarea.



Ilustración 2 Planificación de iteraciones por semana.

#### II.5.2.- Plan de entregas.

La planificación de la entrega de acuerdo a la metodología XP propone agrupar una o varias iteraciones para que el cliente tenga versiones funcionales del sistema. En la ilustración 4 se muestra el Plan de Entregas.

## Capítulo II: Análisis, diseño y construcción de la solución propuesta.



Ilustración 3 Plan de entregas.

### II.6.- Etapa de Diseño.

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

Comenzando una vez que los requisitos de software han sido analizados y especificados, el diseño de software es la primera de tres actividades técnicas (diseño, generación de código y prueba) que se requieren para construir y verificar el software. (Pressman, 2010)

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

## Capítulo II: Análisis, diseño y construcción de la solución propuesta.

### II.6.1.- Prototipo de la interfaz.

En la Ilustración 5 se muestra la interfaz inicial de usuarios y se pueden observar algunas en los anexos.

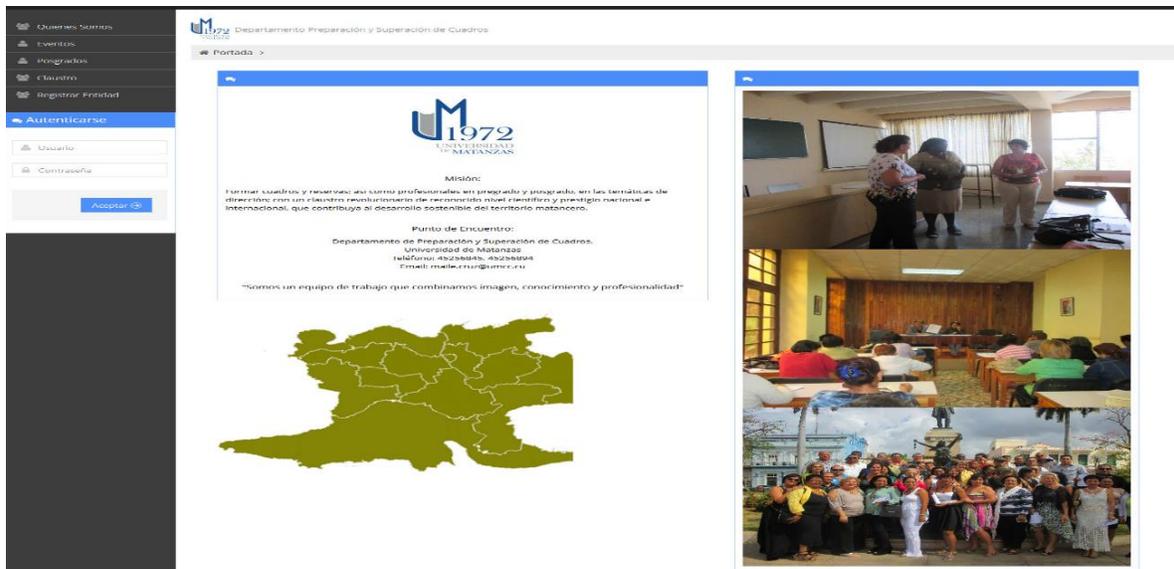


Ilustración 4 Interfaz de usuarios.

### II.6.2.- Metáfora.

Una metáfora es una historia mediante la cual se plasma la arquitectura del sistema, a partir de esta se le da al grupo de desarrolladores una misma visión sobre el proyecto además de brindarles una primera vista muy completa de todas las herramientas que se utilizarán para desarrollar el sistema.

Es muy importante dentro del desarrollo de la metáfora darles nombres adecuados a todos los elementos del sistema constantemente, y que estos correspondan a un sistema de nombres consistente. Esto será de mucha utilidad en fases posteriores del desarrollo para identificar aspectos importantes del sistema.

Se puede decir que la aplicación web permite mejorar la gestión de la información del DPSC.

### II.6.3.- Tareas de ingeniería.

Cada historia de usuario se transformará en Tareas de Ingeniería (TI). Una HU puede transformarse en muchas TI, por esta razón solo se documentarán en este informe las más importantes.

Capítulo II: Análisis, diseño y construcción de la solución propuesta.

En la siguiente tabla 10 se muestran todas las TI a desarrollar en el sistema:

Tabla 10: Tareas de Ingeniería.

| No | Nombre de HU                      | Tareas de ingeniería               | Iteración |
|----|-----------------------------------|------------------------------------|-----------|
| 1  | Diseño y creación de la BD.       | 1-Diseño de la BD.                 | 1         |
|    |                                   | 2-Creación de la BD.               |           |
| 2  | Diseño de la interfaz de usuario. | 3-Diseño de la interfaz principal. |           |
|    |                                   | 4-Diseño de las otras interfaces.  |           |
| 3  | Autenticarse                      | 5-Autenticarse.                    |           |
| 4  | Gestionar usuario                 | 6-Insertar usuario.                | 2         |
|    |                                   | 7-Editar usuario.                  |           |
|    |                                   | 8-Eliminar usuario.                |           |
| 5  | Gestionar profesores              | 9- Insertar profesores.            | 2         |
|    |                                   | 10- Editar profesores.             |           |
|    |                                   | 11-Mostrar profesores.             |           |
|    |                                   | 12- Eliminar profesores.           |           |
| 6  | Gestionar clasificaciones         | 13-Insertar clasificaciones        | 2         |
|    |                                   | 14-Editar clasificaciones          |           |
|    |                                   | 15- Mostrar clasificaciones        |           |
|    |                                   | 16- Eliminar clasificaciones.      |           |
| 7  | Gestionar tipo de posgrado        | 17-Insertar tipo de posgrado.      | 2         |
|    |                                   | 18- Editar tipo de posgrado.       |           |
|    |                                   | 19- Mostrar tipo de posgrado.      |           |
|    |                                   | 20- Eliminar tipo de posgrado.     |           |
| 8  | Gestionar posgrado                | 21-Insertar posgrado.              | 3         |
|    |                                   | 22- Editar posgrado.               |           |
|    |                                   | 23-Mostrar posgrado.               |           |
|    |                                   | 24-Eliminar posgrado.              |           |
| 9  | Insertar participante             | 25-Insertar participante.          | 3         |
| 10 | Gestionar nota                    | 26-Insertar nota.                  | 3         |
|    |                                   | 27- Editar nota.                   |           |
|    |                                   | 28-Mostrar nota.                   |           |
|    |                                   | 29-Eliminar nota.                  |           |
| 11 | Gestionar evento                  | 30-Insertar evento.                | 4         |
|    |                                   | 31- Editar evento.                 |           |
|    |                                   | 32-Mostrar evento.                 |           |
|    |                                   | 33- Eliminar evento.               |           |
|    |                                   | 34-Imprimir evento                 |           |
| 12 | Gestionar asesoria                | 35-Insertar asesoria.              | 4         |
|    |                                   | 36- Editar asesoria.               |           |
|    |                                   | 37-Mostrar asesoria.               |           |
|    |                                   | 38- Eliminar asesoria.             |           |
| 13 | Gestionar tareas                  | 39-Insertar tareas.                | 4         |
|    |                                   | 40- Editar tareas.                 |           |
|    |                                   | 41- Mostrar tareas.                |           |
|    |                                   | 42-Eliminar tareas.                |           |
| 14 | Gestionar entidad                 | 43-Insertar entidad.               |           |
|    |                                   | 44- Editar entidad.                |           |

Capítulo II: Análisis, diseño y construcción de la solución propuesta.

|    |                        |  |   |
|----|------------------------|--|---|
|    |                        | 45-Mostrar entidad.<br>46- Eliminar entidad  | 5 |
| 15 | Gestionar currículum   | 47-Insertar currículum<br>48- Editar currículum<br>49-Mostrar currículum.<br>50- Eliminar currículum                                       | 5 |
| 16 | Gestionar bibliografía | 51-Insertar bibliografía<br>52- Editar bibliografía<br>53-Mostrar bibliografía.<br>54- Eliminar bibliografía<br>55- Descargar bibliografía | 5 |
| 17 | Gestionar archivos     | 56-Insertar archivos<br>57- Editar archivos<br>58-Mostrar archivos<br>59- Eliminar archivos<br>60- Descargar archivos                      | 5 |
| 18 | Gestionar solicitudes  | 61-Insertar archivos<br>62- Editar archivos<br>63-Mostrar archivos<br>64- Eliminar archivos  | 5 |

En las tablas siguientes se muestran se relacionan algunas TI que tenían mayor peso en el desarrollo de esta investigación (ver tablas de la 11 a la 20).

Tabla 11: TI 1: Diseño de la Base de Datos.

| Tareas de Ingeniería   |   |
|--|---|
| <b>Número de Tarea:</b> 1  | <b>Número de Historia de Usuario:</b> 1 |
| <b>Tipo de Tarea:</b> Diseño   | <b>Puntos asignados:</b> 0.5            |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.   |   |
| <b>Descripción:</b> Diseñar la base de datos que permita almacenar toda la información del Departamento. |   |

Capítulo II: Análisis, diseño y construcción de la solución propuesta.

Tabla 12: TI 2: Creación de la Base de Datos.

| Tareas de Ingeniería   |   |
|--|---|
| <b>Número de Tarea:</b> 2  | <b>Número de Historia de Usuario:</b> 1 |
| <b>Tipo de Tarea:</b> Creación   | <b>Puntos asignados:</b> 0.5            |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.   |   |
| <b>Descripción:</b> Crear la base de datos que permita almacenar toda la información del Departamento. |   |

Tabla 13: TI 5: Autenticarse.

| Tareas de Ingeniería  |   |
|---|---|
| <b>Número de Tarea:</b> 5                                   | <b>Número de Historia de Usuario:</b> 3 |
| <b>Tipo de Tarea:</b> Autenticarse                          | <b>Puntos asignados:</b> 1              |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.      |   |
| <b>Descripción:</b> Se autentica el usuario en el sistema . |   |

Tabla 14: TI 6: Insertar Usuario.

| Tareas de Ingeniería   |   |
|--|---|
| <b>Número de Tarea:</b> 6  | <b>Número de Historia de Usuario:</b> 4 |
| <b>Tipo de Tarea:</b> Insertar Usuario   | <b>Puntos asignados:</b> 1              |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.                             |   |
| <b>Descripción:</b> Permite insertar un usuario con los permisos correspondientes. |   |

Tabla 15: TI 9: Insertar profesor.

| Tareas de Ingeniería  |   |
|---|---|
| <b>Número de Tarea:</b> 9   | <b>Número de Historia de Usuario:</b> 5 |
| <b>Tipo de Tarea:</b> Insertar profesor   | <b>Puntos asignados:</b> 1              |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.                              |   |
| <b>Descripción:</b> Permite Insertar un profesor con los permisos correspondientes. |   |

Capítulo II: Análisis, diseño y construcción de la solución propuesta.

Tabla 16: TI 22: Insertar posgrado.

| Tareas de Ingeniería   |   |
|--|---|
| <b>Número de Tarea:</b> 22   | <b>Número de Historia de Usuario:</b> 8 |
| <b>Tipo de Tarea:</b> Insertar posgrado  | <b>Puntos asignados:</b> 1              |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.                             |   |
| <b>Descripción:</b> : Permite insertar posgrado con los permisos correspondientes. |   |

Tabla 17: TI 31: Insertar Evento.

| Tareas de Ingeniería   |  |
|--|--|
| <b>Número de Tarea:</b> 31   | <b>Número de Historia de Usuario:</b> 11 |
| <b>Tipo de Tarea:</b> Insertar evento.   | <b>Puntos asignados:</b> 1               |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.                             |  |
| <b>Descripción:</b> Permite insertar los evento con los permisos correspondientes. |  |

Tabla 18: TI 35: Imprimir Evento.

| Tareas de Ingeniería  |  |
|---|--|
| <b>Número de Tarea:</b> 35  | <b>Número de Historia de Usuario:</b> 11 |
| <b>Tipo de Tarea:</b> Imprimir evento.                                    | <b>Puntos asignados:</b> 0.5             |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.                    |  |
| <b>Descripción:</b> Permite imprimir los eventos después de ser creados . |  |

Tabla 19: TI 37: Editar Asesoría.

| Tareas de Ingeniería   |  |
|--|--|
| <b>Número de Tarea:</b> 37   | <b>Número de Historia de Usuario:</b> 12 |
| <b>Tipo de Tarea:</b> Editar asesoría.   | <b>Puntos asignados:</b> 1               |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.                         |  |
| <b>Descripción:</b> Permite editar asesoría con los permisos correspondientes. |  |

Capítulo II: Análisis, diseño y construcción de la solución propuesta.

Tabla 20: TI 43: Eliminar Tareas.

| <b>Tareas de Ingeniería</b>   |  |
|---|--|
| <b>Número de Tarea:</b> 43  | <b>Número de Historia de Usuario:</b> 13 |
| <b>Tipo de Tarea:</b> Eliminar tareas.  | <b>Puntos asignados:</b> 1               |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.                            |  |
| <b>Descripción:</b> Permite eliminar las tareas correspondientes a las asesorías. |  |

Tabla 21 TI 46: Eliminar entidad.

| <b>Tareas de Ingeniería</b>                               |  |
|---|--|
| <b>Número de Tarea:</b> 46                                | <b>Número de Historia de Usuario:</b> 14 |
| <b>Tipo de Tarea:</b> Eliminar entidad.                   | <b>Puntos asignados:</b> 0.25            |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.    |  |
| <b>Descripción:</b> Permite eliminar todas las entidades. |  |

Tabla 22 TI 47:Insertar currícul.

| <b>Tareas de Ingeniería</b>  |  |
|--|--|
| <b>Número de Tarea:</b> 47   | <b>Número de Historia de Usuario:</b> 15 |
| <b>Tipo de Tarea:</b> Insertar currícul.                               | <b>Puntos asignados:</b> 0.25            |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.                 |  |
| <b>Descripción:</b> Permite insertar los currículos de los profesores. |  |

Tabla 23 TI 52 :Editar bibliografía.

| <b>Tareas de Ingeniería</b>  |  |
|--|--|
| <b>Número de Tarea:</b> 52   | <b>Número de Historia de Usuario:</b> 16 |
| <b>Tipo de Tarea:</b> Editar bibliografía.                         | <b>Puntos asignados:</b> 0.5             |
| <b>Programador responsable:</b> Zuliam Cruz Rodríguez.             |  |
| <b>Descripción:</b> Permite editar la bibliografía de la asesoria. |  |



Capítulo III: Pruebas a la aplicación web de apoyo a la gestión de la información del DPSC.

### III.1 Introducción.

En este capítulo se presentan aspectos sobre las pruebas al software, lo que constituye uno de los pasos más importantes en el diseño e implementación de un sistema. Al finalizar cada iteración se realizan los casos de prueba funcionales y luego se pasa a la siguiente iteración. 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. Obviamente, la mejor forma de que las pruebas estén correctas es incluir al cliente en el diseño de las mismas. Se plantea un estudio de factibilidad y costos. Este capítulo culmina con el análisis de los resultados obtenidos hasta el momento.

### III.2 Pruebas al software.

El único instrumento adecuado para determinar el *status* de la calidad de un producto software es el proceso de pruebas. 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. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante técnicas de prueba.

La fase de pruebas es uno de los elementos que sostiene la metodología XP. En las pruebas de software los procesos soportados por la aplicación se cumplen completamente, es decir, los procesos fluyen desde su inicio hasta el final.

Es importante conocer que existen, según Roger Pressman (2011), principios básicos que guían el buen funcionamiento de las pruebas de software y que es necesario conocerlos antes de aplicar algún método de prueba. Entre ellos se mencionan:

- Que a todas las pruebas se les debería poder hacer un seguimiento hasta los requisitos del cliente.

- Deben planificarse mucho antes de que empiecen.
- Deben comenzar por lo pequeño y progresar hacia lo grande.

La prueba de software se puede definir como una actividad en la cual un sistema o uno de sus componentes se ejecutan en condiciones previamente especificadas, registrándose los resultados obtenidos. Seguidamente se realiza un proceso de evaluación en el que los resultados obtenidos se comparan con los resultados esperados para localizar fallos en el software. (Pressman, 2010)

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

### III.3 Pruebas de Aceptación.

Las Pruebas de Aceptación (PA) las especifica el cliente y se enfocan en las características generales y las funcionalidades de la aplicación. En estas se prueban las funcionalidades exigidas por el cliente, descritas en las TI de las HU que se han implementado. Las pruebas de aceptación se llevan a cabo mediante la redacción de los casos de prueba, teniendo en cuenta el orden de las HU y la prioridad que ha sido asignada a las funcionalidades. Luego se hace la planificación con el cliente de cuándo y cuáles pruebas se llevan a cabo, para así reunir los miembros del proyecto seleccionados para realizarlas. Finalmente, se completa cada uno de los campos de las tablas de las pruebas de aceptación con el resultado de la prueba. Luego de haber superado las pruebas de aceptación podrá considerarse que la aplicación es apta para el uso y despliegue dentro del proyecto.

En las tablas de la 24 a la 31 siguientes aparece una muestra de los casos de prueba desarrollados de las HU en la aplicación.

Tabla 24: PA 3: Autenticarse.

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

Tabla 25: PA 4: Gestionar usuario.

| Pruebas de Aceptación   |               |
|---|---------------|
| Número de Caso de Prueba :4   | No Historia:4 |
| <b>Nombre Caso de Prueba: Test Gestionar usuario.</b>   |               |
| <b>Descripción:</b> Verificar que se gestionen correctamente los usuarios.                              |               |
| <b>Condiciones de ejecución:</b><br>Estar conectado a la Base de Datos                                  |               |
| <b>Entradas:</b><br>Elementos de los usuarios   |               |
| <b>Resultado esperado:</b><br>Se muestren correctamente los usuarios creados, eliminados o modificados. |               |

|  |
|--|
| <p><b>Evaluación:</b></p> <p>Prueba satisfactoria.</p> |
|--|

Tabla 26 PA 5: Gestionar profesor.

| Pruebas de Aceptación   |               |
|---|---------------|
| Número de Caso de Prueba :5   | No Historia:5 |
| Nombre Caso de Prueba: Test Gestionar profesores.                     |               |
| Descripción: Verificar que se gestionen correctamente los profesores. |               |
| Condiciones de ejecución:<br>Estar conectado a la Base de Datos       |               |
| Entradas:<br>Elementos de los profesores.                             |               |
| Resultado esperado:<br>Se gestionan correctamente los profesores.     |               |
| Evaluación:<br>Prueba satisfactoria.                                  |               |

Tabla 27: PA 6: Gestionar clasificación.

| Pruebas de Aceptación  |               |
|--|---------------|
| Número de Caso de Prueba :6  | No Historia:6 |
| Nombre Caso de Prueba: Test Gestionar clasificaciones.                     |               |
| Descripción: Verificar que se gestionen correctamente las clasificaciones. |               |
| Condiciones de ejecución:<br>Estar conectado a la Base de Datos            |               |
| Entradas:<br>Elementos de las clasificaciones.                             |               |

|  |
|--|
| <p><b>Resultado esperado:</b></p> <p>Se gestionan correctamente los clasificaciones.</p> |
| <p><b>Evaluación:</b></p> <p>Prueba satisfactoria.</p>                                   |

Tabla 28: PA 7 Gestionar posgrado.

| Pruebas de Aceptación  |               |
|--|---------------|
| Número de Caso de Prueba :7  | No Historia:7 |
| Nombre Caso de Prueba: Test Gestionar posgrado.                    |               |
| Descripción: Verificar que gestionen correctamente los posgrados . |               |
| Condiciones de ejecución:  |               |
| Estar conectado a la Base de Datos                                 |               |
| Entradas:  |               |
| Elementos de los posgrados .                                       |               |
| Resultado esperado:  |               |
| Se gestionan correctamente los posgrados .                         |               |
| Evaluación:  |               |
| Prueba satisfactoria.  |               |

Tabla 29: PA 10 Gestionar nota .

| Pruebas de Aceptación   |                |
|---|----------------|
| Número de Caso de Prueba :10                                      | No Historia:10 |
| Nombre Caso de Prueba: Test Gestionar nota.                       |                |
| Descripción: Verificar que se gestionen correctamente las notas . |                |
| Condiciones de ejecución:   |                |
| Estar conectado a la Base de Datos                                |                |
| Entradas:   |                |
| Elementos de las notas .  |                |

|  |
|--|
| <b>Resultado esperado:</b><br>Se gestionan correctamente las notas . |
| <b>Evaluación:</b><br>Prueba satisfactoria.                          |

Tabla 30: PA 11 Gestionar evento .

| <b>Pruebas de Aceptación</b>   |                        |
|--|------------------------|
| <b>Número de Caso de Prueba :11</b>  | <b>No Historia:121</b> |
| <b>Nombre Caso de Prueba: Test Gestionar evento.</b>                       |                        |
| <b>Descripción:</b> Verificar que se gestionen correctamente los eventos . |                        |
| <b>Condiciones de ejecución:</b><br>Estar conectado a la Base de Datos     |                        |
| <b>Entradas:</b><br>Elementos de los eventos.                              |                        |
| <b>Resultado esperado:</b><br>Se gestionan correctamente los eventos.      |                        |
| <b>Evaluación:</b><br>Prueba satisfactoria.                                |                        |

Tabla 31 T: PA 12 Gestionar asesoria .

| <b>Pruebas de Aceptación</b>   |                       |
|--|-----------------------|
| <b>Número de Caso de Prueba :12</b>  | <b>No Historia:12</b> |
| <b>Nombre Caso de Prueba: Test Gestionar asesorias.</b>  |                       |
| <b>Descripción:</b> Verificar que se inserten, modifiquen, muestren o eliminen correctamente las asesorias . |                       |
| <b>Condiciones de ejecución:</b><br>Estar conectado a la Base de Datos                                       |                       |
| <b>Entradas:</b><br>Elementos de las asesorias .   |                       |
| <b>Resultado esperado:</b><br>Se gestionan correctamente las asesorias .                                     |                       |
| <b>Evaluación:</b><br>Prueba satisfactoria.  |                       |

#### III.4 Análisis de los resultados obtenidos en las pruebas

Al concluir el desarrollo del proceso de pruebas se lograron resultados satisfactorios en el progreso del sistema informático. Las pruebas fueron desplegadas por cada una de las HU. Los elementos de pruebas abordados, permitieron validar el funcionamiento de la aplicación y los resultados satisfactorios de dichas pruebas.

#### III.5 Estimación del software.

La estimación es el proceso de medición anticipada de la duración, esfuerzos y costes necesarios para realizar todas las actividades y obtener todos los productos asociados a un proyecto. Es necesario tener en cuenta numerosos aspectos que afectan a la estimación como la complejidad del proyecto, su estructuración, el tamaño, los recursos involucrados y los riesgos asociados.

Para la estimación de este proyecto se decide utilizar la fórmula de Bohem, puesto que se adecua en general a los procesos ágiles y como se emplea XP es perfectamente utilizable. La misma plantea que el CT (Costo Total) = CH (Cantidad de Hombres) \* TD (Tiempo de Desarrollo) \* SM (Salario Mensual).

El TD se puede estimar sumando todos los puntos estimados de las HU y agregándole una semana por iteración para las PA. TD es aproximadamente igual a 5 meses.

El SM se puede sustituir por \$500 que es el salario básico aproximado de un ingeniero en Cuba.

La CH es 1, puesto que se trata del autor de este trabajo.

$$CT = CH * TD * SM$$

$$CT = 1 * 7 * \$500$$

$$CT = \$3500$$

Realizando un análisis del costo respecto a los beneficios descritos a lo largo de este trabajo se puede concluir que los aspectos positivos son superiores a los costos. Además, debe señalarse que para esta institución este software resultó gratuito puesto que es el resultado del trabajo de diploma de la autora.

### III.6 Beneficio tangible e intangible.

La aplicación desarrollada presenta beneficios muy favorables ya que este software le ofrece una herramienta para el DPSC, que facilita su trabajo y da solución a la problemática planteada. La aplicación les permite contar con un sistema de fácil acceso y el aumento de la capacidad de almacenamiento de la información permitiendo que de esta forma aumente la calidad de trabajo y por ende disminuye el esfuerzo del usuario para obtener la información necesaria. El usuario tendrá la posibilidad de trabajar con la información de una manera mucho más ágil, satisfaciéndose así las necesidades requeridas por el cliente.

### III.7 Conclusiones parciales del capítulo.

En este capítulo se mostraron los elementos de pruebas de la herramienta y los casos de prueba que se le aplicaron. Las pruebas se convierten en una herramienta de desarrollo y son un paso imprescindible mediante la verificación del buen funcionamiento de un software. La planificación inicial se cumplió por lo que se pudo lograr una aplicación web con funcionalidad y con buena apariencia. Se utilizaron varias de las herramientas más actuales para su desarrollo y el plan de entrega fue cumplido con éxito. Se trató la estimación de los costes y los beneficios.

## Conclusiones

- Se le dio respuesta al problema de investigación puesto que se cumplió el objetivo planteado al desarrollarse un sistema informático que facilita la gestión de la información de algunos procesos que ejecuta el Departamento de Preparación y Superación de Cuadros de la Universidad de Matanzas.
- Se utilizó la metodología XP para el desarrollo de la aplicación web que se presenta, así como las tecnologías de avanzada para la programación de aplicación web, su diseño e implementación.
- Mediante la aplicación de pruebas a la aplicación web, resultó posible obtener resultados favorables, con el consiguiente análisis de errores detectados que fueron subsanados como parte del desarrollo de esta aplicación.
- Se puede afirmar la hipótesis: “si se desarrollara una aplicación web para el DPSC, entonces se facilitaría la gestión de la información de los procesos que ejecuta.

## Recomendaciones

Desde el punto de vista del alcance del presente trabajo y teniendo en cuenta el momento de desarrollo del mismo, se proponen las siguientes recomendaciones:

- Poner en marcha la aplicación y posteriormente realizarle nuevas versiones para que abarque todo el flujo de proceso del DPSC, implementando los restantes procesos.
- Agregarle nuevos reportes que sean de interés para los usuarios que interactúen con la aplicación.

Referencias bibliográficas.

Álvarez, R. (2012). Introducción al HTML. Obtenido de <http://desarrolloweb.com>

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

Betancourt Ávila, J. L. & García Rodríguez, E. 2016. Diagnóstico del Negocio Electrónico de la Universidad de Matanzas. En: Memorias del IV Taller Internacional Las TICs en la Gestión de las Organizaciones, XVI Convención y Feria Internacional Informática 2016, Palacio de Convenciones, La Habana, Cuba (14-18 marzo).

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

Drake, José M. 2008. Programación orientada a objetos: Lenguajes , Metodologías y Herramientas. Santander : Computadoras y tiempo real, 2008.

Cao, L., & otros., y. (2003). Una explicación de la programación extrema(XP). [En línea]

Calero, M. S. (2003). Una explicación de la programación extrema(XP). [En línea]

Cubel, Navarro, y Jose, & María. (2012). Extreme Programming Laboratorio de Sistemas de Información. Valencia, España.

Drake, José M. 2008. Programación orientada a objetos: Lenguajes , Metodologías y Herramientas. Santander : Computadoras y tiempo real, 2008.

Eguiluz, Javier. (2017). Capítulo 1. Introducción. Libros del Web. [En línea] 2017. [Citado el: 8 de Abril de 2017.] [http://librosweb.es/libro/css/capitulo\\_1.html](http://librosweb.es/libro/css/capitulo_1.html)

Falcón, C. (2006). An Epic Review of PyCharm 3. Vim User's Perspective.

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

Fernández, P. (2011). JetBrains anuncia PHP Storm.

Folleto “Código de ética de Cuadros”.

Giraldo, Z. (2006). Herramientas de Ingeniería del Software. 2006.

Gittins, y, R., Hope, & Sian. (2001). A study of Human Solutions in eXtreme Programming. Bangor Reino Unido: Shool of Informatics University of Wales Bango.

Goto, T., Tsuchida, Kensei y Nishino, & Tetsuro. (2014). An Extreme Programming Method for Innovative Software Based on Systems Design and its Practical Study.

Japón: International Journal of Software Engineering & Applications (IJSEA), 2014. Vol. 5, 5.

Highsmith, J. y. C., A. (2001). Agile Software Development: The Business of Innovation. s.l.: IEEE Computer.

Hurtado, Ariel, J., & y Bastiarrica, C. (2005). Modelo de Procesos, Calidad y Mejoramiento. s.l.: Proyecto SIMEP-SW.

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

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

Joskowicz, I. J. (2008). Reglas y Prácticas en Extreme Programming. [En línea].

Lerdorf, Rasmus. 2017. ¿Qué es PHP? Manual de PHP. [En línea] 2017. <http://php.net/manual/es/intro-what-is.php>.

Letelier, Patricio, Penadés, María Carmen y Canós, & H., J. (2006). Metodologías Ágiles en el Desarrollo de Software. Valencia, España: Universidad Politécnica de Valencia.

Luján, M. S. (2001). Clientes Web. s.l.: Editorial Club Universitario.

Martínez, D. (2014). PostgreSQL vs MySQL. Obtenido de <http://danielpecos.com/documents/postgresql-vs-mysql/>

## Referencias bibliográficas.

Marcano, H. (2009). Desarrollo de una aplicación educativa bajo ambiente web, como apoyo para la enseñanza de la asignatura bases de datos orientados a objetos, para la carrera de Licenciatura en Informática de la Universidad de Oriente.

Microsoft. (2017). Microsoft Visual Studio Licensing.

Symfony Documentation. 2016. Symfony en pocas palabras. Libros del Web. [En línea] 2016. [Citado el: 8 de abril de 2017.] [http://librosweb.es/libro/symfony\\_1\\_4/capitulo\\_1/symfony\\_en\\_pocas\\_palabras.html](http://librosweb.es/libro/symfony_1_4/capitulo_1/symfony_en_pocas_palabras.html)

Reynoso, B. C. (2012). Métodos Ágiles en Desarrollo de Software, Introducción a la Arquitectura de Software. Universidad de Buenos Aires.

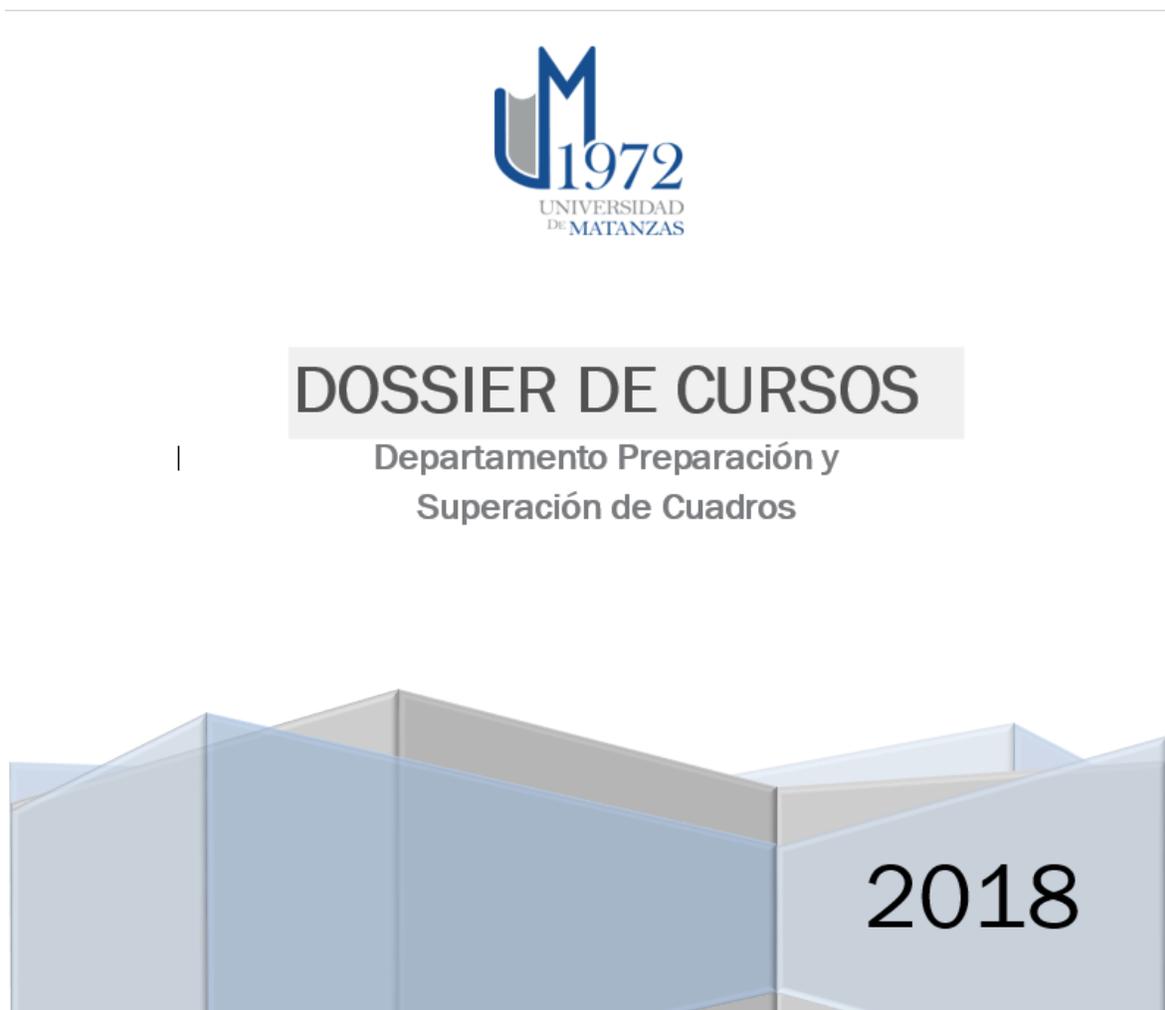
Salgado, M. Dossier del Departamento de Preparación y Superación de Cuadros de la Universidad de Matanzas.

Quintero, E. (2012). Sistema de gestión de la información para la secretaria general de la Universidad de Matanzas "Camilo Cienfuegos". Matanzas.

Valdés, Damián Pérez. (2007). ¿Qué es Javascript? ¿Qué es Javascript? [En línea] 3 de Julio de 2007.

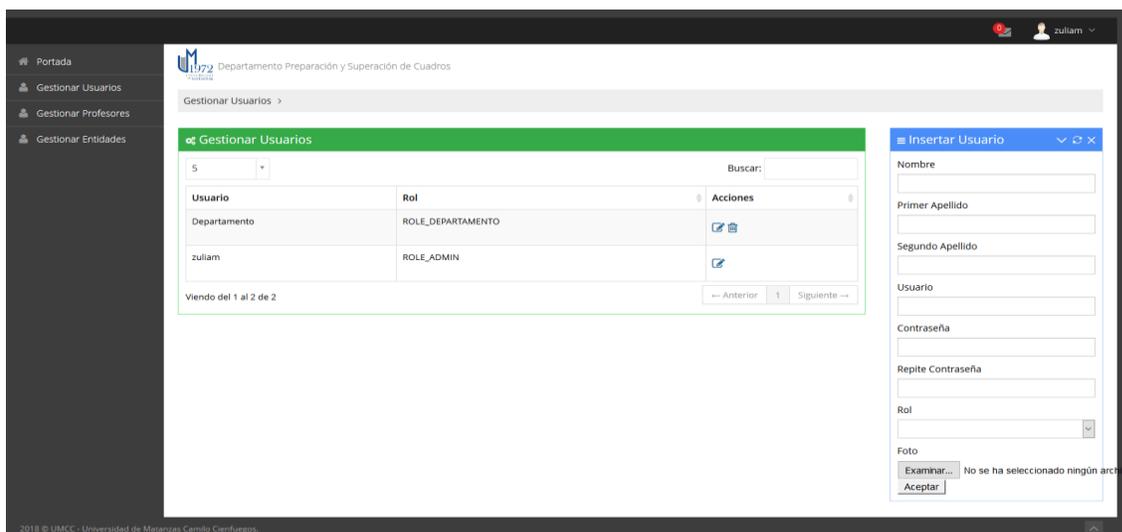
Anexos.

Anexo No. 1: Portada del Dossier del DPSC de la Universidad de Matanzas.





### Anexo No. 3: Interfaz de Gestionar usuario.



### Anexo No. 4: Interfaz de rol Departamento

