

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 1

CENTRO DE IDEOINFORMÁTICA



**Sistema de gestión de ofertas comerciales para
proyectos productivos en la Facultad 1**

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autora

Susana Calás Jerez

Tutores

MSc. Yanedi Abreu Bartomeo

Ing. Adyana Pileta Gammalame

LA HABANA, JUNIO 2019

AÑO 61 DE LA REVOLUCIÓN

« (...) La informática se convertirá en una poderosísima fuerza científica, económica e incluso política del país (...)»

Fidel Castro



DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo Susana Calás Jerez, con carné de identidad 95082246574 declaro ser la autora principal del trabajo titulado “Sistema de Gestión de Ofertas Comerciales para proyectos productivos en la Facultad 1” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

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

Firma del autor

Susana Calás Jerez

Firma del tutor

MSc. Yanedi Abreu Bartomeo

Firma del tutor

Ing. Adyana Pileta Gammalame

DATOS DE LA AUTORA

Nombre y apellidos: Susana Calás Jerez

Correo electrónico: scalas@estudiantes.uci.cu

Institución a la que pertenece: Universidad de las Ciencias Informáticas.

Dirección: Carretera a San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de la Habana, Cuba, Código Postal 19370.

DATOS DE LOS TUTORES

Nombre y apellidos: Yanedi Abreu Bartomeo

Correo electrónico: yabreub@uci.cu

Institución a la que pertenece: Universidad de las Ciencias Informáticas.

Dirección: Carretera a San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de la Habana, Cuba, Código Postal 19370.

Nombre y apellidos: Adyana Pileta Gammalame

Correo electrónico: apileta@uci.cu

Especialidad de graduación: Ingeniero en Ciencias Informáticas.

Institución a la que pertenece: Universidad de las Ciencias Informáticas.

Dirección: Carretera a San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de la Habana, Cuba, Código Postal 19370.

Dedicatoria

A mi hermana Nauria Méndez Trujillo, que, aunque los lazos de sangre no nos unieron, la vida se encargó de hacerlo, para que fuera esa persona especial que junto a mi viviera los momentos de alegrías y tristezas de la universidad y por enseñarme cual es el verdadero significado de la amistad. Nani hoy te dejo ir, para que descanses en paz, pero siempre ten presente que nunca te olvidaré y siempre estarás en mi corazón.

A mi abuela Abelina por ser mi guía, mi ejemplo a seguir, mi segunda madre, por su sacrificio incondicional y por siempre estar para mí. Abuelita siempre serás mi guía y mi ejemplo a seguir. ¡Te Amo!!!

A mis padres Israel y Yamila por darme la vida, por luchar para que saliera adelante y no me faltara nada, por todo su amor y por enseñarme a ser la mujer que soy.

Agradecimientos

A mis abuelos Abelina, Orlando, Silvia Virgen y Manuel por transmitirme durante toda mi vida su sabiduría y por darme tanto amor.

A mis hermanas Aymée y Lianet por ser los tesoros más preciados que tengo en la vida y por darme su apoyo de una forma u otra.

A mis tías Felicia, Graciela y Matilde por estar siempre presente en mi vida y por querer siempre lo mejor para mí.

A mi primo Asley Ballester Calás por ser mi guía, por sus consejos y por apoyarme en mis decisiones.

A mis primos Yuliet Sánchez Jerez y Yulián Sánchez Jerez por su cariño, su amistad, los consejos y valores que me ha transmitido para que no cometiera sus mismos errores y fuera quien soy el día de hoy.

A mi familia de manera general por todo su apoyo incondicional durante toda mi vida.

A mis tutoras Adyana y Yanedi por su apoyo incondicional durante el desarrollo de este trabajo de diploma, lo cual no habría sido posible sin ellas.

A los miembros del tribunal Dolly, Yeneit y Aylín por su excelente labor como tribunal.

A los profesores que durante estos 5 años de la carrera me han transmitido sus conocimientos y han sido un ejemplo a seguir.

A Yojahny Chávez Marrero por la comprensión, el cariño y el apoyo en los días más difíciles durante estos 5 años de la universidad.

A Mayra de la O Barrientos y Yailén Esmeralda Salgado Reyes por su amistad y su apoyo incondicional y por ser mis pingüinas.

A Leonardo Novoa Rodríguez por ser uno de mis mejores amigos en estos 5 años de la universidad, por ser paciente y por sacarme una sonrisa en los momentos de gran estrés.

A Geidis Sánchez Michel por la ayuda incondicional que me ha dado durante estos años de la universidad.

A mi tata Reiner Blanco Crespo por ser mi gran amigo y mi pilar en todo momento, por aguantar mis malos momentos y por estar siempre a mi lado.

A Tymeé Yero Trotman por ser mi amiga desde los primeros momentos de nuestras vidas hasta el día de hoy y por estar junto a mí en los buenos y malos momentos.

A María Elena Olazabal Álvarez por brindarme su amistad y cariño desde el primer momento en que la conocí, y por apoyarme en los momentos más difíciles.

A Reimer D. Malleza Romero por su apoyo incondicional durante estos últimos meses, por ser mi consejero, mi paño de lágrimas y el hombre amigo que siempre te aguanta para no caer en el camino.

A Yoan Alomá Sotolongo por su amistad y por estar siempre dispuesto a ayudarme en los buenos y malos momentos.

A mi prima Danelys Claudia Jerez Yero por estar siempre en los buenos y malos momentos.

A mis amistades del antiguo 410: Adrián Castillo, Luis Alberto Justiz, Yadriel, Maybel, Daniel y Lisbeth.

A personitas que se han vuelto muy importantes en mi vida: Tania, Claudia Rafaela, Eliwis, Ortelio, Lidice, Ariel y Leisy.

A personas muy especiales que conocí y me han apoyado en mi 5to año: Merelia, María Elena Fornari, Jonathan, Yagmín, Iris María Suárez, Yanay Hacobet y Anthony.

A mis amistades que durante toda mi vida hemos vivido momentos especiales y me han dado todo su apoyo: Aianys, Yunisleidis, Roxana Ros, Marién, Melisa, Carolina, Yania, Pablo, Jose, Harol, Manuel, Yenis, Josué, Darisnelis, Milena, Yaquelín, Aylén, Lurdes, Arlet, Adrián Carballo, Andy, Guillermo, Pedro Iván, Pedro Javier, Reimer, Shadet, Jose Carlos, Ivette de Torres, Liz Camila, Hymet (La yes), Danesa, Marisdelis y Elianis González.

A Fidel Castro Ruz y la Revolución por darme la oportunidad de estudiar en esta universidad y permitirme ser quien soy.

RESUMEN

La presente investigación consistió en desarrollar un sistema para la Facultad 1 que permite la gestión de ofertas comerciales. El sistema posee determinadas características que no se encuentran en herramientas similares analizadas, y que fueron tomadas como muestra, siendo denominadas como sistemas homólogos. Para la implementación se utilizó el entorno de desarrollo integrado NetBeans, como Sistema Gestor de Bases de Datos (SGBD) MySQL, Visual Paradigm como herramienta de modelado, para la implementación Drupal 7.43, PHP 7 y CSS3 y se eligió como servidor web Apache 2.4. Para guiar todo el proceso de desarrollo se usó la metodología AUP en su personalización para la UCI. Las pruebas de software arrojaron que la propuesta de solución es funcional, segura y con un rendimiento adecuado. La consulta con expertos permitió validar que la propuesta de solución contribuye a la mejora del tiempo y el esfuerzo en la gestión de ofertas comerciales.

Palabras clave: Base de cálculo, Drupal 7, ficha técnica, gestión de ofertas comerciales.

Contenido

INTRODUCCIÓN.....	1
CAPÍTULO 1: “Herramientas para la gestión de ofertas comerciales, funcionalidades y tecnologías para su desarrollo”.....	6
1.1 Conceptos fundamentales.....	6
1.1.1 Gestión de Ofertas Comerciales	6
1.1.2 Tiempo.....	6
1.1.3 Esfuerzo	¡Error! Marcador no definido.
1.2 Estudio de Sistemas Homólogos	6
1.3 Sistema Gestor de Contenido.....	8
1.3.1 Sistema Gestor de Contenido Drupal 7	9
1.4 Análisis de las tecnologías, metodología y herramientas a utilizar.....	11
1.4.1 Metodología de desarrollo de software	11
1.4.1.1 Metodología de desarrollo Variación de AUP para la UCI.....	11
1.4.2 Tecnologías de desarrollo	12
1.5 Conclusiones parciales	18
CAPÍTULO 2 “Caracterización del sistema de gestión de ofertas comerciales para proyectos en la Facultad 1”	19
2.1 Características de la propuesta de solución.....	19
2.2 Modelo de dominio	19
2.2.1 Levantamiento de la información	21
2.2.2 Fuentes para la obtención de requisitos	21
2.2.3 Técnica de identificación de requisitos.....	21
2.2.4 Especificación de requisitos de software.....	22
2.2.5 Descripción de requisitos de software.....	25
2.2.6 Validación de requisitos de software	26
2.3 Análisis y diseño	27
2.3.1 Descripción de la arquitectura de software y los patrones de diseño	27
2.4 Modelo de diseño.....	29
2.4.1 Diagrama de clases.....	29
2.4.2 Patrones de diseño.....	30
2.5 Diagrama de despliegue	32
2.6 Diagrama de datos.....	32
2.7 Conclusiones parciales	34

3.1 Modelo de implementación.....	35
3.2 Estándares de codificación utilizados.....	36
3.4 Plan de iteraciones	40
3.4 Pruebas de software.....	41
3.5 Aplicación de las pruebas de software	41
3.5.1 Pruebas Funcionales	42
Resultados de las pruebas por caso de pruebas	45
3.5.2 Pruebas de seguridad	46
3.5.3 Pruebas de rendimiento.....	46
3.5.4 Pruebas de aceptación	48
3.6 Validación de la Hipótesis.....	49
3.7 Interfaz principal	52
Bibliografía.....	¡Error! Marcador no definido.

Índice de tablas

Tabla 1: Requisitos funcionales del sistema (elaboración propia).....	23
Tabla 2 Historia de usuario autenticar usuario (elaboración propia)	26
Tabla 3 Tiempo de duración de las iteraciones (elaboración propia).....	41
Tabla 4 Resultados de aplicar la prueba de caja negra (elaboración propia).....	43
Tabla 5 Resultados de las pruebas por caso de pruebas (elaboración propia).....	46
Tabla 6 Resultados de la aplicación de la herramienta Acunetix WVS v8 para pruebas de seguridad (elaboración propia).....	46
Tabla 7 Caso de prueba de aceptación para la historia de usuario "Crear ficha técnica"(elaboración propia)	48
Tabla 8 Expertos seleccionados para realizar la validación de la hipótesis (elaboración propia).....	50
Tabla 9 Distribución de frecuencias a partir de datos primarios (elaboración propia)	51
Tabla 10 Historia de usuario asignar rol.....	58
Tabla 11 Historia de usuario crear roles.....	59
Tabla 12 : Historia de usuario mostrar rol.....	59
Tabla 13 Historia de usuario modificar rol	60
Tabla 14 Historia de usuario eliminar rol	60
Tabla 15 Historia de usuario crear ficha técnica	60
Tabla 16 Historia de usuario mostrar ficha técnica.....	61
Tabla 17 Historia de usuario modificar ficha técnica	61
Tabla 18 Historia de usuario eliminar ficha técnica	62
Tabla 19 Historia de usuario crear base de cálculo.....	62
Tabla 20 Historia de usuario mostrar base de cálculo	63
Tabla 21 Historia de usuario modificar base de cálculo.....	63
Tabla 22 : Historia de usuario eliminar base de cálculo.....	64
Tabla 23 Historia de usuario exportar ficha técnica.....	64
Tabla 24 Historia de usuario exportar base de cálculo.	¡Error! Marcador no definido.
Tabla 25 Instrumento de validación del sistema de gestión de ofertas comerciales	65

Índice de figuras

Figura 1 : Diagrama de clases del Modelo de Dominio (elaboración propia)	20
Figura 2 Arquitectura de la propuesta de solución	28
Figura 3 Diagrama del diseño de la HU autenticar usuario (elaboración propia)	30
Figura 4 Diagrama de despliegue (elaboración propia)	32
Figura 5 Diagrama de Modelo de Datos (elaboración propia)	33
Figura 6 Diagrama de componentes (elaboración propia)	36
Figura 7 Ejemplo de uso de los operadores en el archivo update.php	37
Figura 8 Ejemplo de uso de la estructura de control if en el archivo update.php	37
Figura 9 Ejemplo de uso de la estructura de control switch en el archivo update.php	38
Figura 10 Ejemplo de declaración de un array en el archivo update.php	38
Figura 11 Ejemplo de uso de las comillas simples en el archivo update.php	39
Figura 12 Ejemplo de uso de las comillas dobles en el archivo update.php	39
Figura 13 Ejemplo de uso del punto y coma en el archivo update.php	39
Figura 14 Ejemplo de nombres de las funciones en el archivo update.php	39
Figura 15 Ejemplo de nombres de variables en el archivo update.php	40
Figura 16 Grafo resultante de del método caja blanca camino básico (elaboración propia)	44
Figura 17 Resultados de las pruebas de aceptación (Fuente: elaboración propia)	49
Figura 18 Resultados obtenidos de la validación de hipótesis (elaboración propia)	51
Figura 19 Interfaz del sistema	52

INTRODUCCIÓN

La gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo (Ameijide, 2016). Según el Instituto de Gestión de Proyectos (*Project Management Institute, PMI*), los procesos se guían por cinco etapas: iniciación, planificación, ejecución, control y cierre. La gestión del proyecto se puede aplicar a casi cualquier tipo de proyecto y es ampliamente utilizado para controlar los complejos procesos de los proyectos de desarrollo de software incluyendo el proceso de marketing que según (Rattinger, 2018) es una de las disciplinas con mayor campo de acción en el mundo profesional. Es el conjunto de principios enfocados a entender las necesidades de un público con la finalidad de impulsar las ventas y llega hasta la forma adecuada de comunicación para crear un vínculo con las personas, que les aporten un valor agregado.

Con la visión de futuro del Comandante en Jefe Fidel Castro Ruz, surge la Universidad de Ciencias Informáticas (UCI) en el año 2002. Fidel, como estrategia fundacional, recomendó que la universidad fuese concebida como un centro de nuevo tipo, de alcance nacional, atípico y con tareas concretas en el proyecto de informatización de la sociedad cubana, con énfasis en la producción de software (Historia_Universidad de las Ciencias Informáticas, 2018).

La UCI se encuentra dividida en seis facultades con 15 centros productivos que desarrollan actividades de I+D+i, encargados de las aplicaciones informáticas, el desarrollo tecnológico y las investigaciones asociadas. De conjunto, conforman una red de trabajo colaborativo que opera bajo normas y procedimientos comunes, posibilitando la reutilización de componentes y eficiencia industrial. Los centros de desarrollo de software realizan más de 200 proyectos al año, con gran impacto en la informatización de la sociedad cubana, en áreas como la seguridad ciudadana, administración pública, industria, educación, gestión empresarial, salud y biotecnología, entre otras (Centros de Desarrollo_Universidad de las Ciencias Informáticas, 2018).

En la Facultad 1 se encuentran tres centros productivos, el Centro de Ideoinformática (CIDI), el Centro de Identificación y Seguridad Digital (CISED) y el Centro de Software Libre (CESOL) que presentan como una de sus funciones el desarrollo de los procesos de producción, formación, investigación, mercadotecnia y colaboración, a través de una estrategia encaminada al logro de los objetivos (Centros de Desarrollo_Universidad de las Ciencias Informáticas, 2018).

Dentro de estos centros entre los procesos que se desarrollan se encuentra, la gestión de proyectos y a su vez la misma enmarca a la gestión de ofertas comerciales, que es el proceso donde una vez identificada una oportunidad de negocio, y evaluadas las posibilidades de la empresa, es necesario mostrar al cliente nuestro interés y capacidad para la correcta ejecución de los trabajos. Para ello es necesario abordar la fase oferta, en la cual se prepara la documentación que le permite al cliente juzgar la idoneidad del ofertante para la realización de los trabajos, así como fijar el precio de los mismos (Benítez, 2011).

La gestión y procesamiento de la información referida a las ofertas comerciales manejadas por la UCI, se realiza manualmente a través de fichas técnicas y bases de cálculo generadas por el Asesor de Mercadotecnia de cada centro productivo de la universidad. Donde las fichas técnicas son el documento enviado como respuesta a una solicitud de un cliente, en el que como mínimo se indica el precio y las condiciones de entrega del producto y las bases de cálculo son necesarias para evitar dudas respecto de los días de desarrollo de un proyecto, para el cálculo de los gastos y el presupuesto que se necesita para el desarrollo del proyecto.

Este es un proceso complicado y puede inducir a cometer errores humanos, pues se realizan manualmente, generan costos en tiempo, presupuesto y de medios de trabajo al tener que replicar informaciones, volver a imprimir documentos o malgastar tiempo innecesariamente. Actualmente existe en la Universidad un sistema para la gestión de proyecto conocido por sus siglas como GESPRO, pero este no genera dichas fichas técnicas y bases de cálculo correspondientes, lo que provoca que el proceso sea engorroso y lento para el Asesor de Mercadotecnia, teniendo en cuenta que se pueden recibir varios proyectos en el mismo mes en cada centro productivo.

Por lo anteriormente planteado se define como **problema de investigación**: ¿Cómo disminuir el tiempo en la gestión de ofertas comerciales de los proyectos que se ejecutan en los centros productivos asociados a la Facultad 1?

Objeto de Estudio de la presente investigación lo constituye el proceso de gestión de la información referente al área de Mercadotecnia.

Se plantea como **objetivo general**: Desarrollar un sistema utilizando Drupal 7 para disminuir el tiempo empleado en la gestión de ofertas comerciales de los proyectos que se ejecutan en los centros productivos asociados a la Facultad 1.

Como **objetivos específicos** se tienen:

1. Sistematizar los referentes teóricos relacionados con el tiempo y esfuerzo para la gestión de ofertas comerciales de los proyectos productivos asociados a la Facultad 1 de la UCI.

2. Diseñar las funcionalidades del sistema para la gestión de ofertas comerciales de los proyectos asociados a la Facultad 1 de la UCI.
3. Implementar las funcionalidades del sistema para la gestión de ofertas comerciales de los proyectos asociados a la Facultad 1 de la UCI.
4. Validar el sistema para la gestión de ofertas comerciales de los proyectos asociados a la Facultad 1 de la UCI.

Se selecciona como **campo de acción** el proceso de gestión de ofertas comerciales para proyectos en la Facultad 1.

La investigación presenta la siguiente **Hipótesis**: un sistema de gestión de ofertas comerciales para proyectos productivos disminuirá el tiempo empleado en el proceso de gestión y procesamiento de la información referida a las ofertas comerciales manejadas por Facultad 1.

Se define como **variable independiente**: un sistema de ofertas comerciales, el mismo consiste en una aplicación informática que contendrá un conjunto de funcionalidades relacionadas con la gestión y procesamiento de la información referida a las ofertas comerciales manejadas por los centros productivos asociados a la Facultad 1 de la UCI.

Como **variable dependiente** se plantea la disminución de tiempo y esfuerzo en el proceso de gestión y procesamiento de la información.

Operacionalización de las variables

Variable	Dimensiones	Indicadores	Sub-Indicadores	Escala
Sistema de Gestión de Ofertas Comerciales	UCI	Usabilidad	Visibilidad del Sistema	(%)
			Lenguaje común entre sistema y usuario	(%)
Disminución de tiempo en el proceso de gestión de la información.	Asesores de Mercadotecnia	Tempo del proceso de configuración	Alta	1-5(seg)
			Media	6-10(seg)
			Baja	11-15(seg)

Tareas de investigación:

1. Realización de la sistematización de los referentes teóricos relacionados con el tiempo y esfuerzo para la gestión de ofertas comerciales de los proyectos que se ejecutan en los centros productivos asociados a la Facultad 1 de la UCI.
2. Descripción del estado del arte de las tecnologías utilizadas y de las candidatas a utilizar.
3. Levantamiento de información.
4. Selección de las tecnologías, herramientas y estándares que se necesitan para implementar la propuesta de solución.
5. Selección de patrones de diseño y arquitectura.
6. Implementación de la solución
7. Diseño de pruebas de software.
8. Validación de la solución.

Los métodos científicos que se utilizan en la presente investigación son:

Métodos teóricos:

- **Histórico-Lógico:** Para analizar el desarrollo del proceso de gestión de ofertas comerciales a través del tiempo de proyectos informáticos de este tipo (Sistemas de Gestión de Ofertas Comerciales).
- **Analítico-Sintético:** Para seleccionar los elementos más importantes relacionados con la gestión de ofertas comerciales.

Métodos empíricos:

- **Entrevista:** Usado durante el intercambio con el cliente para adquirir información sobre el negocio y los requisitos que se deben cumplir en el desarrollo del sistema.
- **Encuesta:** Utilizado para la recopilación de información en la aceptación por los usuarios.

La tesis consta de la presente introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos que ampliarán la información que se aporta en la investigación.

Capítulo 1: “Herramientas para la gestión de ofertas comerciales, funcionalidades y tecnologías para su desarrollo”. Contiene los fundamentos teóricos que sustentan al sistema de gestión de ofertas comerciales. En el mismo se hace un estudio de sistemas homólogos analizando el objetivo, su funcionamiento y tecnologías que utilizan. Además, se describen los principales conceptos para comprender el dominio del problema, así como las tendencias, metodologías, tecnologías y herramientas que deben ser empleadas en la implementación de la propuesta de solución.

Capítulo 2: “Caracterización del sistema de gestión de ofertas comerciales para proyectos que se ejecutan en los centros productivos en la Facultad 1”. En este capítulo se explica cómo se desarrolla el flujo actual de los procesos y se describe la propuesta de solución para resolver el problema planteado. Por otra parte, se especifican los requisitos funcionales y no funcionales y los elementos fundamentales del diseño y de la arquitectura que se deben tener en cuenta para el desarrollo del sistema, ayudado por la realización de diagramas.

Capítulo 3: “Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1”. En este capítulo se incluye la programación realizada a partir de los requisitos, así como las pruebas de software realizadas para su validación. Se establecen los estándares de codificación que se tuvieron en cuenta para el desarrollo del sistema.

CAPÍTULO 1: “Herramientas para la gestión de ofertas comerciales, funcionalidades y tecnologías para su desarrollo”

En el presente capítulo se define el marco conceptual que sustenta la investigación estrechamente vinculada a las herramientas de gestión de ofertas comerciales. Se exponen los principales conceptos relacionados con el estado del arte de las herramientas que permiten la gestión de ofertas comerciales. A partir de los resultados se analizan diferentes lenguajes y herramientas que sean factibles para desarrollar una solución a los inconvenientes planteados, así como la metodología para guiar el proceso de desarrollo de software.

1.1 Conceptos fundamentales

Para lograr un mejor entendimiento del problema a investigar se tuvieron en cuenta conceptos importantes tales como: gestión de ofertas comerciales, sistema de gestión, tiempo y mercadotecnia o marketing lo que permite una mejor comprensión del objeto de estudio.

1.1.1 Gestión de Ofertas Comerciales

Es el proceso de documentación de un proyecto y es la matriz diseñada para gestionar las políticas, procedimientos y procesos del desarrollo de un proyecto.

1.1.2 Sistema de gestión

Son programas diseñados para manejar las políticas y los procedimientos de una organización de manera eficaz. Este mecanismo de gestión documenta cada uno de los procesos de la empresa, aunque cada industria tiene sus propias necesidades (Siigo, 2019).

1.1.3 Tiempo

Magnitud física que permite ordenar la secuencia de los sucesos, estableciendo un pasado, un presente y un futuro, y cuya unidad en el sistema internacional es el segundo (Real Academia Española, 2018).

1.1.4 Mercadotecnia o Marketing

Es un proceso social y administrativo, que implica un intercambio entre grupos e individuos de una sociedad para satisfacer mutuas necesidades y en donde el producto es su variable básica, pues engloba tanto a los bienes como a los servicios que comercializa una empresa (Sánchez, y otros, 2017).

1.2 Estudio de Sistemas Homólogos

A continuación, se realiza un análisis de distintas herramientas utilizadas para la gestión de ofertas comerciales. Se tienen en cuenta los sistemas creados para la gestión de ofertas y más utilizados en la actualidad, se exponen las características, ventajas y desventajas para realizar una valoración crítica.

1.2.1 Sistema de gestión comercial(Insoft)

Permite la generación automática de facturas a partir de uno de dos criterios: pedidos con o sin promociones. El sistema permite que el usuario configure este concepto en el tipo de pedido. De este sistema se toma la forma de configurar el tipo del pedido que el caso de la propuesta de solución se adapta a la forma en que se selecciona el tipo de proyecto, así como sus generalidades.

1.2.2 Sistema de gestión Distribución Comercial Mayorista

Permite generar y configurar sus ofertas y pedidos, gestionar su seguimiento, reutilizar las ofertas total o parcialmente, generar los pedidos de forma manual y automática. Utiliza la movilidad para el envío de sus pedidos en su red de negocio. Sistema de Gestión que cuenta con todos los procesos de negocio para la distribución comercial mayorista. No sólo es un software, es una transferencia del conocimiento sectorial. De este sistema se toma la forma de generar los pedidos automáticamente adaptándola a las necesidades de la propuesta de solución.

1.2.3 GESTIONPRO

Promete generar su facturación por ventas en forma fácil, rápida y segura, así como facilitar el control de cuentas corrientes de clientes y proveedores, cheques, bancos y vendedores. Permite obtener más de 190 reportes con información impositiva e histórica por mercadería, servicios, compras y ventas. Aplicable para venta de servicios, distribuidoras, ventas mayoristas y minoristas y productos de fabricación propia. Compatible con impresoras fiscales segunda generación EPSON TM-T900FA, HASAR SMH/PT-1000F, HASAR SMH/PT-250F y SAM4S ELLIX-40F. De este sistema se toma la forma de guardado y consulta de los datos históricos, teniendo en cuenta que en la propuesta de solución es necesario mostrar las fichas técnicas y bases de cálculo que se han realizado anteriormente.

Valoración crítica de los sistemas estudiados

Luego de realizar un análisis de las características de los sistemas antes mencionados se obtuvo que realizan funciones como: crear, eliminar, actualizar, modificar y generar automáticamente un documento. Estas funciones son necesarias para la propuesta de solución y fueron tomadas como referencia para implementar las del sistema, pero estos sistemas no responden a las necesidades de desarrollo de la UCI, ya que son *software* privativos o de pago.

1.3 Sistema Gestor de Contenido

Un sistema gestor de contenido (CMS) es un paquete de software que ofrece herramientas para la creación, edición y gestión de contenido en un sitio o aplicación web. Los CMS proveen la habilidad de administrar la estructura de un sitio, la apariencia de las páginas publicadas y la navegación dedicada a los usuarios (*Vasundhara, 2015*).

Un CMS ofrece una serie de características que simplifican el proceso de construcción, implementación y administración de sitios web entre ellas se puede encontrar las siguientes:

- Una interfaz administrativa.
- Un repositorio de base de datos de contenidos.
- Un mecanismo para asociar la información que se almacena en la base de datos con una página en el sitio web.
- Un conjunto de herramientas para la creación, publicación y gestión de contenidos.
- Un componente para la creación y gestión de menús y elementos de navegación.
- Las herramientas necesarias para definir y aplicar temas.
- Gestión de usuarios.
- Un marco de seguridad.
- Capacidades web 2.0, tales como foros, blogs, encuestas.
- Taxonomía y etiquetado.
- Formularios en línea con capacidades de comercio electrónico (*Tolmison, 2010*).

1.3.1 Sistema Gestor de Contenido Drupal 7

Drupal es un CMS de código abierto, altamente modular con énfasis en la colaboración. Contiene funcionalidades básicas en su núcleo y se puede agregar funcionalidades a través de la instalación de módulos. También separa con éxito la gestión de contenido de la presentación de contenidos. Es distribuido bajo los términos de la Licencia Pública General (GNU/GPL). Drupal es también un marco de gestión de contenido (CMF). Además de proporcionar herramientas para la construcción de sitios web, ofrece vías para que programadores y desarrolladores lo personalicen usando módulos (**Byron y Berry, 2012**).

Características Principales:

- Sistema de módulos flexible: Los módulos son una característica de Drupal que pueden modificar y agregar funcionalidades en un sitio web.
- Sistema de temas personalizable: Las salidas en Drupal son totalmente personalizables.
- Sistema de contenido extensible y de entidades: Se pueden definir nuevos tipos de contenido (*blogs, eventos, noticias*) y agregar campos para distintos tipos de contenido.
- Permisos de acceso basado en Roles: Cada usuario en Drupal contiene uno o varios roles con privilegios determinados.
- Herramientas para publicaciones y colaboraciones: Drupal contiene soportes para herramientas como comentarios, foros, y perfiles de usuarios personalizados (**Sheltren, Newton y Catchpole, 2014**).

Ventajas:

- Drupal es un software de código abierto con una comunidad de más de 648000 usuarios y más de 10000 desarrolladores.
- Soporta más de 55 idiomas.
- El sitio oficial de Drupal lista alrededor de 12101 módulos y temas gratuitos para extender sus funcionalidades.
- Cientos de universidades del mundo utilizan Drupal para el desarrollo de sitios web.

- Algunos de los sitios más potentes del mundo utilizan esta poderosa, flexible y escalable tecnología (**Pandiyan y Singhal, 2015**).

Seguridad:

- Actualizaciones de seguridad del núcleo y los módulos.
- Comunidad de seguridad activa
- Soporte de php 7.
- Módulos para mejorar la seguridad, por ejemplo, *captcha*.
- Posee un validador de fortaleza de contraseña para que sea más segura.

Usabilidad:

- Los enlaces administrativos para editar los elementos de las páginas existentes están disponibles en todas las páginas web, sin tener que ir a una página de administración primero.
- Presenta pestañas verticales, un componente de interfaz reutilizable que ofrece resúmenes automáticos y aumenta la facilidad de uso.
- Presenta soporte para usos de horario.

Rendimiento:

- Soporte a APC.
- Integración con memcache
- Compresión de páginas con gzip
- Momificación de css y js
- Sistema de cache nativa.
- Campos de datos personalizados se pueden unir a los nodos, usuarios, comentarios y términos de la taxonomía (Drupal 7, 2015).

1.4 Análisis de las tecnologías, metodología y herramientas a utilizar.

En el desarrollo de sistemas en Drupal 7 se utilizan varias tecnologías como lenguajes de programación, herramientas de modelado, gestores de bases de datos, y servidores web. A continuación, se realiza una descripción de las tecnologías y herramientas utilizadas para la implementación de la propuesta de solución y se selecciona la metodología de desarrollo que se utiliza para guiar el proceso de desarrollo del *software*.

1.4.1 Metodología de desarrollo de software

Una metodología de desarrollo de software describe un entorno que es usado para organizar, planificar, y dirigir un proceso de desarrollo de un software. Existen varias metodologías de desarrollo de software, todas contienen algunas etapas básicas del ciclo de desarrollo de software como son la planificación, análisis, diseño, implementación y mantenimiento (Verma, Bansal y Pandey, 2015).

1.4.1.1 Metodología de desarrollo Variación de AUP para la UCI

En el desarrollo de la propuesta de solución se utiliza la metodología de desarrollo de software Variación de AUP para la UCI en su escenario 4, una variante realizada por la Universidad de las Ciencias Informáticas a la metodología ágil AUP (*Agile Unified Process*) y definida por la universidad como guía para la actividad productiva ya permite estandarizar los diferentes productos de trabajo que se generan en sus centros productivos y se adapta al ciclo de vida definido para los proyectos de la UCI (Rodríguez, 2015). Esta metodología propone las siguientes fases:

Inicio: Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo, costo y decidir si se ejecuta o no el proyecto.

Ejecución: En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elaboran la arquitectura, el diseño, se implementa y se libera el producto. Durante esta fase el software es transferido al ambiente de los usuarios finales o entregado al cliente junto con la documentación. Además, en esta transición se capacita a los usuarios finales sobre la utilización de la aplicación.

Cierre: En el cierre se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

El desarrollo de la investigación se centrará en la fase de Ejecución y transitará por las siguientes disciplinas: requisitos, análisis, diseño, implementación, pruebas interna y pruebas de aceptación, definidas en la metodología. Porque la fase de inicio se basa en la gestión de proyecto, estimaciones de tiempo, esfuerzo, costo y decidir si se ejecuta o no el proyecto, en la fase de cierre se analizan tantos los resultados del proyecto, como su ejecución y las actividades formales de cierre del proyecto que no se realizan en la presente investigación.

1.4.2 Tecnologías de desarrollo

Para el desarrollo de sistemas de gestión en Drupal 7 y la modelación de los artefactos de la metodología seleccionada se utilizan diversas tecnologías, a continuación, se describen y caracterizan cada una de las utilizadas en el desarrollo de la propuesta de solución.

1.4.2.1 Lenguajes utilizados

Pre-Procesador de Hipertexto (PHP 7)

El preprocesador de hipertexto (PHP) es un lenguaje de programación del lado del servidor, es poderoso y fácil de usar. Presenta las características de la sintaxis de los lenguajes C, Java y Perl, es ampliamente utilizado en el mundo para el desarrollo web. PHP tiene una licencia de código abierto esto significa que los desarrolladores de PHP puedan descargar el código fuente del mismo y hacer los cambios pertinentes (Chávez, 2016).

Lenguaje marcado de hipertexto (HTML 5)

Es un lenguaje de marcado de hipertexto para presentar y estructurar el contenido en internet. Es la quinta revisión y nueva versión del estándar HTML. HTML5 ofrece nuevas características que proporcionan no sólo un rico soporte multimedia (vídeo y audio), sino que también mejoran el apoyo para la creación de aplicaciones web. Esta nueva tecnología puede proporcionar una solución rentable para implementar aplicaciones para apoyar diferentes dispositivos (por ejemplo, ordenador, tableta, teléfono inteligente personal) sin tener que construir aplicaciones diferentes para cada tipo de dispositivos.

Características

- Semántica: para describir el contenido con mayor precisión.
- Conectividad: para comunicarse con el servidor en formas nuevas e innovadoras.
- Multimedia: para proporcionar apoyo a los contenidos de vídeo y audio.
- Gráficos: gráficos 3D en 2D y Efectos.
- Rendimiento e Integración: proporcionar optimización de la velocidad y un mejor uso del hardware (Cahyadi y Bandung, 2014).

Hojas de Estilo en Cascada (CSS 3)

Las Hojas de Estilo en Cascada (CSS) es un lenguaje utilizado para describir el aspecto y el formato de los documentos HTML. CSS ha sido ampliamente adoptado en la web y la práctica del desarrollo móvil, ya que permite una separación limpia del contenido de la presentación (Tsantalís, 2016). La separación de los contenidos y su presentación presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados “documentos semánticos”). Según (Paganotti, 2015) esta separación mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. Las nuevas características de CSS3 respecto a sus versiones anteriores incluyen soporte para selectores adicionales, sombras, esquinas redondeadas, múltiples fondos, animaciones y transparencia (W3C, 2018).

Lenguaje Unificado de Modelado (UML)

UML (Unified Modeling Language, 2015) fue adoptado como estándar del *Object Management Group* (OMG) en 1997 debido a que representa una colección de las mejores prácticas de ingeniería que han sido probadas con éxito en el modelado de sistemas. Es un lenguaje para la especificación, visualización, construcción y documentación de sistemas, no solo de software.

1.4.2.2 Sistemas gestores de bases de datos

Un Sistema Gestor de Bases de Datos (DBMS) es una colección de programas que permiten a los usuarios crear y mantener una base de datos. Sistema software de propósito general que facilita los procesos de definición, construcción y manipulación de la base de datos para distintas aplicaciones.

- Definición de las bases de datos: especificar tipos de datos, estructuras y restricciones.

- Construcción de las bases de datos: almacenar datos.
- Manipulación de las bases de datos: consultar, actualizar el diseño y generar informes.

Si la base de datos son los datos almacenados el DBMS es el programa o conjunto de programas que gestionan y mantienen consistentes estos datos (Garzón, 2014).

Drupal soporta diferentes gestores de bases de datos, esto se consigue mediante una capa de abstracción de la base de datos que convierte las instrucciones genéricas proporcionadas por Drupal en instrucciones particulares de cada base de datos.

MySQL 5.7

MySQL es una base de datos que trabaja con lenguajes de script. Es considerado como de mediana escala y es adecuado para aplicaciones web de tamaño medio. MySQL permite que varios usuarios a través de múltiples hilos puedan acceder a datos relacionales. MySQL soporta múltiples herramientas para la gestión. Una de las herramientas populares la herramienta de administración web phpMyAdmin escrito en PHP.

La popularidad de MySQL es debido a su flexibilidad, el rendimiento sólido y funciones útiles. MySQL se puede desplegar en múltiples sistemas operativos, incluyendo las plataformas Windows, Mac OS X y muchas distribuciones de Linux.

También cuenta con las interfaces de programación de aplicaciones preparadas para los lenguajes de programación más populares como: C, C ++, Java, Perl, PHP y Ruby (Kemppainen, 2015). Es un sistema de gestión de bases de datos relacional, licenciado bajo GNU/GPL. Fue diseñado para proveer un veloz Método de Acceso Secuencial Indexado, conocido como ISAM (*Indexed Sequential Access Method*).

Este tipo de carga de datos, caracterizado por la ejecución de consultas cortas, combinado con técnicas como el cacheo de consultas, ayuda a mejorar su desempeño. Por estas características se selecciona MySQL en su versión 5.7 para el desarrollo de la propuesta de solución.

1.4.2.3 Servidores de aplicaciones web

Un servidor web es un programa que utiliza HTTP (Protocolo de Transferencia de Hipertexto) para brindar los archivos que forman las páginas web a los usuarios, en respuesta a sus peticiones, que

se envían por los clientes HTTP de sus equipos. El servidor web se encarga de interpretar el código PHP, generar la página HTML correspondiente y entregársela al usuario que la ha solicitado a través de su navegador.

El proceso es un ejemplo del modelo de cliente / servidor. Todos los equipos que brinden servicios web deben tener programas de servidor web. Los principales servidores web son Apache (el servidor web más instalado ampliamente), *Internet Information Server de Microsoft* (IIS) y Nginx. Otros servidores web incluyen servidor NetWare de Novell, Google Web Server (GWS) y la familia de servidores dominio de IBM (What is, 2015).

Nginx

Nginx es un servidor web y proxy¹ libre, de código abierto y de alto rendimiento, ofrece estabilidad, un gran conjunto de características básicas de los servidores web, configuración sencilla, y bajo consumo de recursos (Nginx, 2015). Según (Kholodkov, 2015) Nginx se integra con diferentes tecnologías web como Python y PHP y presenta soporte de una amplia y activa comunidad.

Haciendo uso de sockets asíncronos utiliza un proceso por núcleo para manejar miles de conexiones, lo que permite un consumo de carga de la CPU y la memoria mucho más ligera (Nedelcu, 2014).

Apache

Apache web Server es el servidor HTTP con mayor participación en el mercado mundial. Apache se caracteriza por ser estable, modular, tener código abierto y ser gratuito. Además, es altamente configurable de acuerdo a las necesidades de la organización que lo utilice. Apache registra en archivos log toda la operación con el fin de hacer más fácil la tarea del administrador suministrando información útil para la toma de decisiones, por ejemplo, el ajuste en una directiva de configuración. Por otra parte, Apache ofrece módulos especializados en distintas actividades, por ejemplo, integración con lenguajes de programación en el lado del cliente, módulos de seguridad y módulos de redirección.

¹ Es un programa o dispositivo, consiste en interceptar las conexiones de red que un cliente hace a un servidor de destino.

El uso de estos módulos puede ser aprovechado para dar un mejor servicio a sus usuarios sin comprometer a la organización que expone sus contenidos en una red pública como internet para ser recuperados a través de solicitudes HTTP realizadas por un navegador. Apache se caracteriza también por ser multiplataforma, lo que permite su ejecución en la mayoría de sistemas operativos, tales como Unix, GNU/Linux y Windows (Uribe, 2014).

El sitio oficial de Drupal recomienda utilizar Apache como servidor web de los proyectos creados en Drupal, la mayoría de los módulos establecen en su implementación a este servidor web por defecto, esto provoca incompatibilidad en algunos casos con otros servidores web, obligando a cambiar la configuración del módulo para su correcto funcionamiento. Por estas características se selecciona Apache en su versión 2.4 para el desarrollo de la propuesta de solución y Nginx para su despliegue.

1.4.2.4 Herramientas CASE

Se puede definir a las herramientas de Ingeniería de Software Asistida por Ordenador (CASE por sus siglas en inglés) como cualquier herramienta que se utiliza para automatizar alguna actividad (Bareisa, 2015). Existen variadas herramientas CASE dentro de las que se encuentran Erwin, Rational Rose, Visual Paradigm, estas herramientas ayudan en las tareas relacionadas con la fase de desarrollo del software como la especificación, estructurado, análisis, diseño, codificación, pruebas y otras actividades como gestión de proyectos y gestión de la configuración. Estas están destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste en términos de tiempo y de dinero.

Visual Paradigm for UML 8.0

Visual Paradigm for UML es una herramienta CASE que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir diagramas de diversos tipos, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML (Visual Paradigm, 2016).

1.4.2.5 Herramienta de desarrollo

NetBeans 8.0

NetBeans es un entorno de desarrollo integrado y una plataforma de desarrollo. Aunque inicialmente, sólo podía ser utilizado para desarrollar aplicaciones Java, a partir de la versión 6, NetBeans soporta varios lenguajes de programación, ya sea a través de una función de apoyo, o mediante la instalación de complementos adicionales, algunos de estos lenguajes son Java, C, C ++, PHP, HTML, JavaScript. Groov y Scala (Heffelfinger, 2015). NetBeans proporciona funcionalidades adicionales como la compilación en tiempo real, la comprobación de tipos, refactorización, navegadores de clase y soluciones rápidas para los errores en tiempo de compilación.

1.4.2.6 Herramientas para realización de pruebas

Apache Jmeter

La aplicación Jmeter es un software de código abierto, creada en java y diseñada para probar el comportamiento funcional de un sitio web y medir el rendimiento. Fue diseñado originalmente para pruebas de aplicaciones web, pero desde entonces se ha expandido a otras funciones de prueba. Jmeter puede ser utilizado para probar el rendimiento tanto en recursos estáticos y dinámicos, lenguajes dinámicos como PHP, Java, ASP.NET, objetos Java, bases de datos y consultas. Se puede utilizar para simular una carga pesada en un servidor, grupo de servidores, la red o el objeto para probar su resistencia o para analizar el rendimiento general bajo diferentes tipos de carga (Apache Jmeter, 2015).

Acunetix Web Vulnerability Scanner

Acunetix Web *Vulnerability Scanner* es una herramienta para realizar pruebas de seguridad en aplicaciones web. Acunetix ha sido pionera en la aplicación web de tecnología de análisis de seguridad. Sus ingenieros se han centrado en la seguridad web desde 1997 y ha desarrollado una ingeniería líder en el análisis de sitios web y detección de vulnerabilidades (Acunetix, 2016).

Algunas de sus características son:

- Las herramientas de testeado de inyección SQL y de Crosssite scripting más avanzadas y profundas de la industria
- Herramientas avanzadas de penetración, como HTTP Editor y HTTP Fuzzer
- Herramientas para fácil aseguramiento de formularios web y contraseñas

- Soporte para páginas con captcha, single sign-on y mecanismos con factor de autenticación
- Facilidad de generación de informes amplios, incluyendo informes de cumplimiento PCI
- El escaneo inteligente detecta el tipo de servidor web y lenguaje de la aplicación
- Acunetix escanea y analiza sitios web incluyendo contenido flash, SOAP y AJAX
- Permite explorar un servidor web y ejecutar comprobaciones de seguridad contra los servicios de red que se ejecutan en el servidor.

1.5 Conclusiones parciales

El estudio de los sistemas homólogos permitió conocer las principales características y las diferentes valoraciones acerca de los sistemas de ofertas comerciales. A partir del análisis de las diferentes herramientas y tendencias para los sistemas de gestión de ofertas comerciales, se determinaron las características que constituyen la base para el diseño de las funcionalidades que se definen en la propuesta de solución. El estudio sobre las metodologías, herramientas y lenguajes permitió definir los componentes base para el desarrollo de la solución, donde se define a AUP-UCI como metodología de desarrollo y como herramienta CASE Visual Paradigm 8.0 para el modelado de los artefactos del análisis y diseño de la solución. Se recurrió al Visual Paradigm para modelar con la cual se podrá realizar el análisis y diseño de la solución a través del lenguaje UML. Para la implementación se utilizará el lenguaje de programación php, la herramienta NetBeans IDE en su versión 8, el servidor web Apache en su versión 2.

CAPÍTULO 2 “Caracterización del sistema de gestión de ofertas comerciales para proyectos en la Facultad 1”

En el presente capítulo se caracteriza la propuesta de solución a través de la definición de los requisitos funcionales y no funcionales, las tareas a realizar durante la implementación, la arquitectura y patrones utilizados en el desarrollo del sistema de gestión de ofertas comerciales teniendo en cuenta que estuviesen presentes las buenas prácticas del diseño y la programación. Además, se presentan los principales artefactos generados en las primeras fases del desarrollo.

2.1 Características de la propuesta de solución

En vistas de dar solución a los objetivos expuestos y teniendo en cuenta las necesidades de mejorar el tiempo y minimizar el esfuerzo en la gestión de ofertas se propone el desarrollo de un sistema de gestión de ofertas comerciales de forma que el asesor de mercadotecnia pueda realizar su trabajo ágilmente y sin generar muchos gastos. En la solución existirá un espacio para gestionar las fichas técnicas y bases de datos asociadas a cada proyecto, además de permitir la eliminación, actualización y reutilización de las mismas en proyectos similares.

El sistema debe ser multiplataforma, adaptable a distintas resoluciones y estándar e independiente del gestor de base de datos a utilizar, además debe ser compatible con diferentes navegadores web como Firefox, Opera y Chrome.

2.1.1 Roles y permisos

En el sistema de gestión de ofertas comerciales intervienen dos roles principales:

Usuario Autenticado: Podrá configurar el sistema de gestión de ofertas comerciales si presenta los permisos para ello.

Usuario Administrador: Podrá establecer los permisos de utilización del sistema y la configuración del mismo.

2.2 Modelo de dominio

Para la modelación del entorno y los procesos que intervienen en el sistema a informatizar se utiliza el modelo de dominio, en el cual se representan los conceptos más importantes y significativos en el desarrollo de la propuesta de solución. Este modelo permite identificar las relaciones entre entidades, clases conceptos comprendidos en el ámbito del problema, al igual que sus atributos y

CAPÍTULO 2. Caracterización del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

restricciones (García, 2017). Lo antes mencionado no implica que se modelan clases o componentes de softwares, si no componentes propiamente dichos.

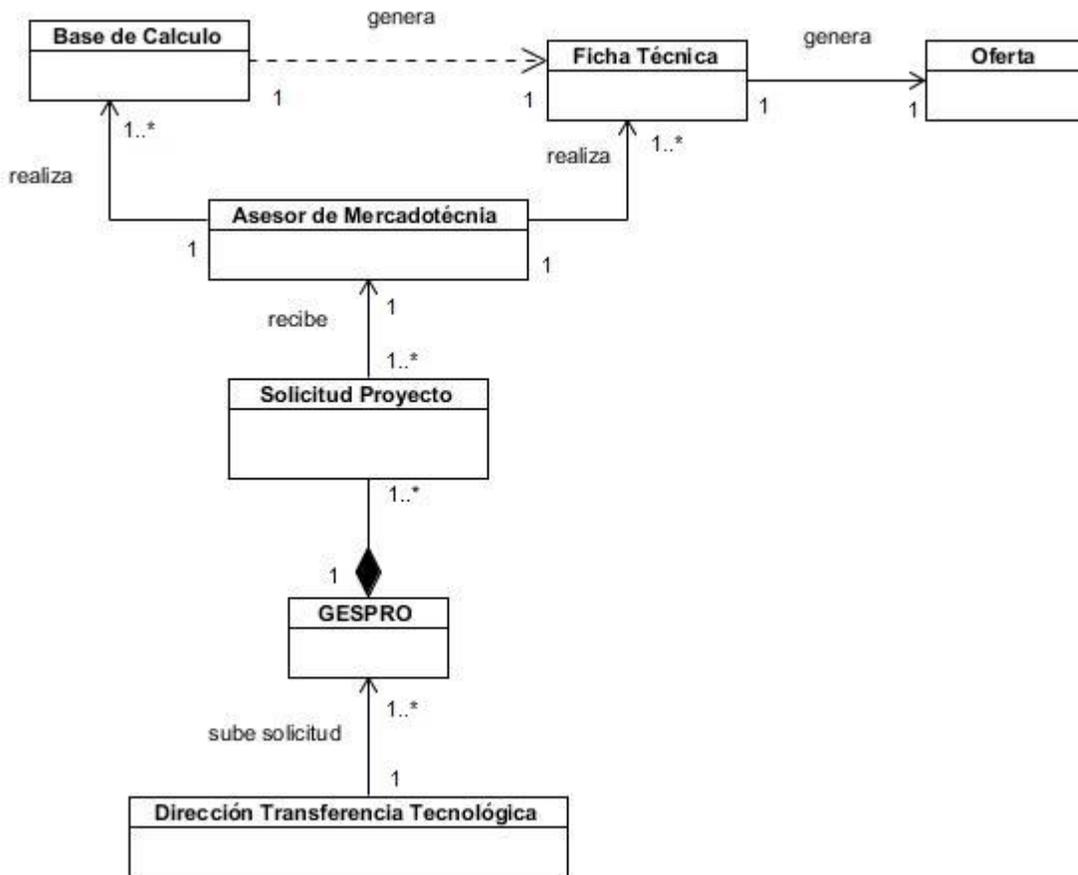


Figura 1 : Diagrama de clases del Modelo de Dominio (elaboración propia)

GESPRO: sistema de gestión de proyectos utilizado por la universidad para gestionar cada uno de los proyectos que en esta se realizan.

Solicitud de proyecto: solicitud que sube al GESPRO la Dirección de Transferencia Tecnológica de un nuevo proyecto.

Asesor de Mercadotecnia: usuario autenticado que realiza la oferta utilizando el sistema de gestión de ofertas comerciales.

Base de Cálculo: documento realizado para evitar dudas respecto de los días de desarrollo de un para el cálculo de los gastos y el presupuesto que se necesita para el desarrollo del proyecto y necesario para realizar la ficha técnica.

CAPÍTULO 2. Caracterización del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

Ficha Técnica: documento enviado como respuesta a una solicitud de un cliente en el que como mínimo se indica el precio y las condiciones de entrega del producto. Se considera una propuesta de contrato.

Oferta: Oferta generada por el sistema, resultado final.

2.2.1 Levantamiento de la información

En esta etapa es necesario hacer uso de una identificación exhaustiva y correcta de los requisitos funcionales y no funcionales de la propuesta de solución. Los requisitos definen qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades y las restricciones que se imponen. Los requisitos funcionales (RF) son las condiciones que el sistema debe cumplir, mientras que los requisitos no funcionales (RNF) son cualidades que el producto debe tener (Chávez, 2016).

2.2.2 Fuentes para la obtención de requisitos

Las fuentes de obtención de requisitos utilizadas fueron:

- Propuesta de solución del sistema de gestión de ofertas comerciales.
- Análisis de los sistemas homólogos (Ver epígrafe 1.2)
- Asesores de Mercadotecnia de CIDI, CESOL, CISED.

2.2.3 Técnica de identificación de requisitos

Las técnicas de identificación de requisitos de *software* permiten identificar las necesidades de negocio de clientes y usuarios. Son mecanismos que se utilizan para recolectar la información necesaria en la obtención de los requisitos de una aplicación, permiten investigar aspectos generales para posteriormente ser especificados con un mayor detalle, requieren ser adecuadamente orientadas para cubrir la información que se requiere capturar (Pressman, 2015).

Entrevista

Se utilizó como técnica de extracción de requisitos la entrevista, que es la más tradicional de las técnicas de obtención y consiste en reuniones analista-interesado en las cuales suceden preguntas y respuestas para extraer el dominio de la aplicación (Pressman, 2016). La forma en que se realizó esta técnica fue: el autor de la investigación se entrevistó con el cliente, y a partir de la entrevista realizada detectaron 16 requisitos funcionales y 9 requisitos no funcionales (*Anexo 1*).

CAPÍTULO 2. Caracterización del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

2.2.4 Especificación de requisitos de software

Una especificación de requerimientos de software (ERS) es un documento que se crea cuando debe especificarse una descripción detallada de todos los aspectos del software que se va a elaborar (Pressman, 2016). A continuación, se describen los requisitos funcionales y no funcionales de la propuesta de solución.

Requisitos funcionales

Los requisitos funcionales (RF) son declaraciones de los servicios que debe proporcionar el sistema, describe lo que este debe hacer. En la tabla 1 se especifican los requisitos funcionales que debe cumplir la propuesta de solución.

No.	Nombre	Descripción	Prioridad	Complejidad
RF1	Autenticar Usuario	El sistema deberá permitir la autenticación de los usuarios.	Alta	Baja
RF2	Asignar roles	El sistema permitirá asignar los roles y permisos a cada usuario para la utilización del sistema.	Alta	Baja
RF3	Crear rol	El sistema permitirá crear un rol que dependiendo de los permisos podrá utilizar el sistema.	Alta	Baja
RF4	Mostrar rol	El sistema permitirá mostrar el rol que anteriormente fue creado.	Alta	Baja
RF5	Modificar rol	El sistema permitirá modificar el rol que anteriormente fue creado.	Alta	Baja
RF6	Eliminar rol	El sistema permitirá eliminar el rol que anteriormente fue creado.	Alta	Baja
RF7	Crear ficha técnica	El sistema permitirá crear las fichas técnicas.	Alta	Alta

CAPÍTULO 2. Caracterización del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

RF8	Mostrar ficha técnica	El sistema permitirá mostrar las fichas técnicas.	Alta	Alta
RF9	Modificar ficha técnica	El sistema permitirá modificar las fichas técnicas.	Alta	Alta
RF10	Eliminar ficha técnica	El sistema permitirá eliminar las fichas técnicas.	Alta	Alta
RF11	Crear base de cálculo	El sistema permitirá crear las base de cálculos que servirán para crear las fichas técnicas correspondientes.	Alta	Alta
RF12	Mostrar base de cálculo	El sistema permitirá mostrar las base de cálculos que servirán para crear las fichas técnicas correspondientes.	Alta	Alta
RF13	Modificar base de cálculo	El sistema permitirá modificar las base de cálculos que servirán para crear las fichas técnicas correspondientes.	Alta	Alta
RF14	Eliminar base de cálculo	El sistema permitirá eliminar las base de cálculos que servirán para crear las fichas técnicas correspondientes.	Alta	Alta
RF15	Exportar ficha técnica	El sistema permitirá exportar las fichas técnicas que serán realizadas a partir de las bases de cálculos correspondientes.	Alta	Media

Tabla 1: Requisitos funcionales del sistema (elaboración propia).

Requisitos no funcionales

Los requisitos no funcionales son aquellos requisitos que no describen información a guardar, ni funciones a realizar, sino que son propiedades que hacen al producto usable, rápido o confiable.

CAPÍTULO 2. Caracterización del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

Además, se conocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el software (Sommerville , 2015).

RNF1: Interfaz

RNF1.1: El menú de configuración del sistema presentará una interfaz agradable, utilizando colores claros, en tonalidades violetas y lilas teniendo en cuenta los utilizados por la universidad en la marca de mercado XEDRO.

RNF1.2: Los textos serán presentados en tipografías compatibles con varios sistemas operativos y se resaltarán lo encabezados en negrita.

RNF1.3: El diseño será fluido colocando los componentes de configuración en lugares apropiados para que el usuario pueda utilizarlo de forma rápida.

RNF2: Usabilidad

RNF2.1: Se utilizarán tamaños de letra apropiados para personas con alguna discapacidad visual.

RNF3: Rendimiento

RNF3.1: El sistema debe responder en un máximo de 3 segundos las solicitudes de los usuarios.

RNF3.2: El sistema debe soportar un mínimo de 500 peticiones.

RNF3.3: El sistema debe ser escalable, permitiendo incorporarle nuevas funcionalidades sin afectar las existentes.

RNF3.4: El sistema deberá continuar su funcionamiento bajo circunstancias anormales, para ello debe ser hospedado en un servidor dedicado, con tolerancia a fallos.

RNF4: Operacionales

RNF4.1: Para la distribución del producto y su instalación se necesitará un entorno de trabajo compuesto por:

- Servidor Web: Apache.
- Base de Datos: MySQL 5
- Sistema gestor de contenido: Drupal 7.65.

CAPÍTULO 2. Caracterización del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

- Lenguaje de programación: PHP 7 o versiones posteriores recomendadas.

RNF5: Mantenimiento y soporte

RNF5.1: Las operaciones de mantenimiento del sistema no deberán exceder de 2 días.

RNF6: Software

RNF6.1: El sistema funcionará en un entorno de trabajo compuesto por Navegadores web Chrome 72, Firefox 44, Opera 34 e Internet Explorer a partir de su versión 10.

RNF7: Seguridad

RNF7.1: El sistema será utilizado solo por un usuario administrador y un usuario registrado con permisos.

RNF7.2: Los errores deben mostrar la menor cantidad de detalles posible, para evitar brindar información que comprometa la seguridad e integridad del sistema.

RNF8: Legales

RNF8.1: Se utilizará la licencia GNU/GPL versión 2 para el CMS Drupal.

RNF8.2: Se utilizará la licencia PHP License.

RNF8.3: Se utilizará la licencia de Visual Paradigm 8.

RNF9: Hardware

RNF9.1: El servidor de aplicaciones web y de base de datos deben poseer como mínimo un CPU Dual Core a 2.20 GHz con 4 Gb de RAM.

RNF9.2: El servidor de base de datos debe poseer una capacidad mínima de 10 GB.

RNF9.3: El servidor de aplicaciones web debe poseer una capacidad mínima de 10 GB.

2.2.5 Descripción de requisitos de software

La descripción de los requisitos funcionales de la propuesta de solución se realiza mediante las historias de usuario. Este producto de trabajo permite describir las funcionalidades que el sistema debe poseer, sean requisitos funcionales o no funcionales. A continuación, se muestra la historia de usuario del RF1 que preside de una prioridad alta, las restantes historias de usuario se mostrarán en el anexo 2.

CAPÍTULO 2. Caracterización del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

Historia de usuario	
Número: 1	Nombre Historia de usuario: Autenticar usuario
Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 1 día
Riesgo en Desarrollo: Bajo	Tiempo Real: 2 días
Descripción: El sistema deberá permitir la autenticación de los usuarios.	
Observaciones: El sistema mostrará 2 campos, usuario (campo de texto) obligatorio y contraseña (campo tipo password) obligatorio. El usuario insertará su nombre de usuario, ejemplo (admin) y su contraseña, ejemplo (junio2019), luego al presionar el botón iniciar sesión entrará en el sistema con los permisos correspondientes.	
Prototipo de interfaz:	

Tabla 2 Historia de usuario autenticar usuario (elaboración propia)

2.2.6 Validación de requisitos de software

La validación de los requerimientos analiza la especificación a fin de garantizar que todos ellos han sido enunciados sin ambigüedades; que se detectaron y corrigieron las inconsistencias, las omisiones y los errores, y que los productos del trabajo se presentan conforme a los estándares establecidos para el producto (Pressman, 2016). A continuación, se describen las técnicas de validación de requisitos utilizadas en la propuesta de solución:

CAPÍTULO 2. Caracterización del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

Revisión Técnica Formal

Una revisión técnica formal (RTF) es una actividad del control de calidad del software realizada por ingenieros de software. Los objetivos de una RTF son: descubrir los errores en funcionamiento, lógica o implementación de cualquier representación del software; verificar el cumplimiento de sus requisitos; garantizar su representación de acuerdo a los estándares predefinidos; obtener uniformidad en su desarrollo y hacer proyectos más manejables (Pressman, 2016).

Diseño de casos de pruebas

Los diseños de casos de pruebas permiten crear un conjunto de entradas y salidas esperadas que sean efectivos para descubrir defectos en los programas y muestren que el sistema satisface sus requerimientos. Para su diseño, se selecciona una característica del sistema o componente que se está probando, un conjunto de entradas que ejecutan dicha característica, se documentan las salidas esperadas o rasgos de salida y donde sea posible se diseña una prueba automatizada que demuestre que las salidas reales y las esperadas son las mismas.

2.3 Análisis y diseño

La disciplina del análisis tiene como propósito conseguir una comprensión más precisa de los requisitos, refinarlos y estructurarlos; el diseño tiene el propósito de formular los modelos que se centran en los requisitos no funcionales y en el dominio de la solución y que prepara para la implementación y prueba del sistema.

2.3.1 Descripción de la arquitectura de software y los patrones de diseño

Analizando que una arquitectura de software es de mucha importancia, ya que a un sistema le proporciona un impacto directo sobre la capacidad de este para satisfacer lo que se conoce como los atributos de calidad del sistema y que los patrones de diseño son una guía para resolver problemas comunes en programación, se describe la arquitectura de software y los patrones de diseños utilizados en el desarrollo de la propuesta de solución (Sommerville , 2015).

2.3.2 Arquitectura de software

Dentro de los principales componentes de la arquitectura de Drupal están: núcleo, entidades, tipos de contenido, módulos, bloques, menús, temas, permisos de usuarios, plantillas. Este gestor de contenido utiliza para su funcionamiento el patrón arquitectónico N-Capas. Drupal está estructurado por temas de interfaces de interacción para con los usuarios, los cuales se pueden descargar de Internet o simplemente crearlos como plantillas en PHP, HTML y CSS. Posee una capa de

CAPÍTULO 2. Caracterización del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

abstracción de base de datos implementada y soportada para MySQL, aunque puede añadir soporte para varias bases de datos. Los módulos que contienen algoritmos en PHP que proporcionan el funcionamiento del sistema, actúan como páginas servidoras y separan la interfaz gráfica de la información. Esta arquitectura tiene como objetivo principal separar los diferentes aspectos del desarrollo mediante capas y permitir intercambiar porciones de la aplicación sin necesidad de modificarla completamente, es adaptable producto al soporte de adición de módulos, que son al final porciones de la aplicación que se modifican, sin tener que realizar cambios en el resto de la aplicación (Suárez, 2016).

Drupal estructura los contenidos en una serie de elementos básicos como: los nodos, módulos, bloques y menús, permisos de usuario y plantillas como se muestra a continuación.

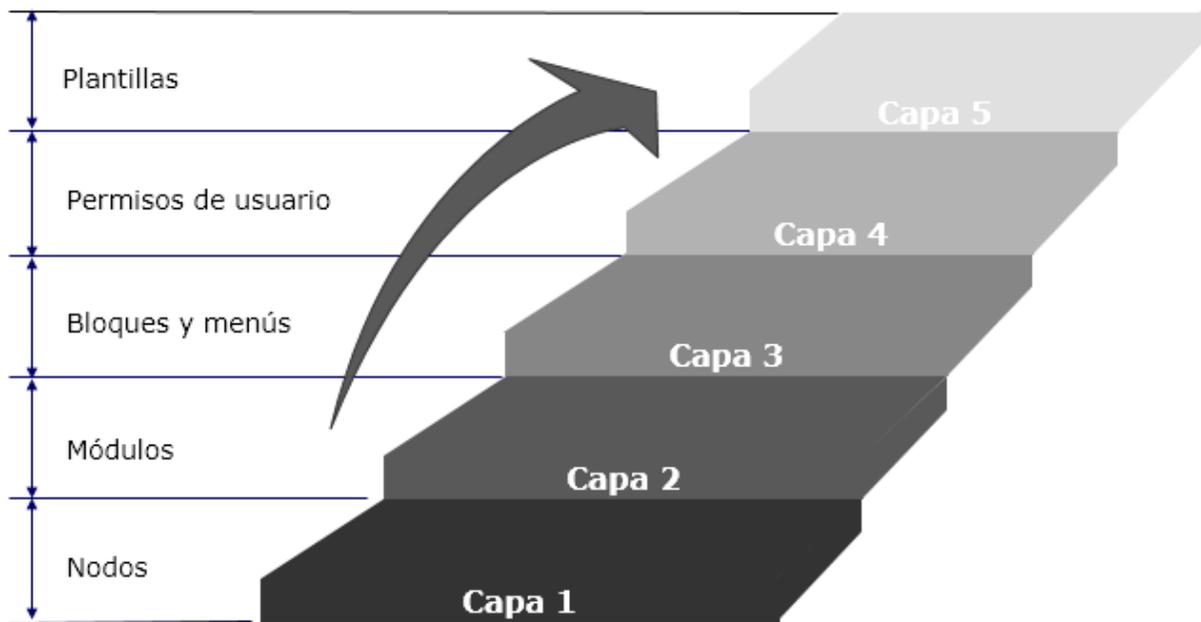


Figura 2 Arquitectura de la propuesta de solución (Elaboración propia)

- La primera capa representa la información almacenada en los Nodos (tipos de contenido a publicar ejemplo noticias, avisos, adquisiciones).
- En la segunda se encuentran los Módulos que son utilizados en la aplicación (ejemplos los módulos *ctool*, *views*, *jcarousel*), que son los elementos que operan sobre los Nodos y otorgan funcionalidad a Drupal permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada sitio web.

CAPÍTULO 2. Caracterización del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

- La tercera esta descrita por los Bloques y Menús que permiten estructurar y organizar los contenidos en la página web (ejemplo: los menús ¿Quiénes Somos?, Enlaces y bloque de adquisiciones, Video).
- La cuarta recoge los Permisos de Usuarios donde Drupal dispone de un registro de usuarios y de roles que permiten especificar qué tareas pueden realizar y a qué contenidos puede acceder cada tipo de usuario (ejemplos: roles de administrador, editor y usuario).
- La quinta y última capa maneja las Plantillas y es la que establece la apariencia gráfica o estilo de la información que se le muestra al usuario (ejemplo: el tema biblioteca).

2.4 Modelo de diseño

El modelo de diseño es planteado como un modelo de objetos que describe la realización física de los casos de usos, centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar, constituyendo una entrada principal en la actividad de implementación (López Martínez , 2015).

2.4.1 Diagrama de clases

Los diagramas de clases (DC) son un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos; las cuales pueden ser asociativas, de herencia, de uso y de contenido. Expresan la estructura u organización del software en términos de clases. Además, constituyen el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer, como para mostrar cómo puede ser construido.

Para el diseño de la propuesta de solución fueron generadas un total de 16 diagramas de clases, a continuación, se muestra el diagrama del diseño de la HU autenticar usuario.

CAPÍTULO 2. Caracterización del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

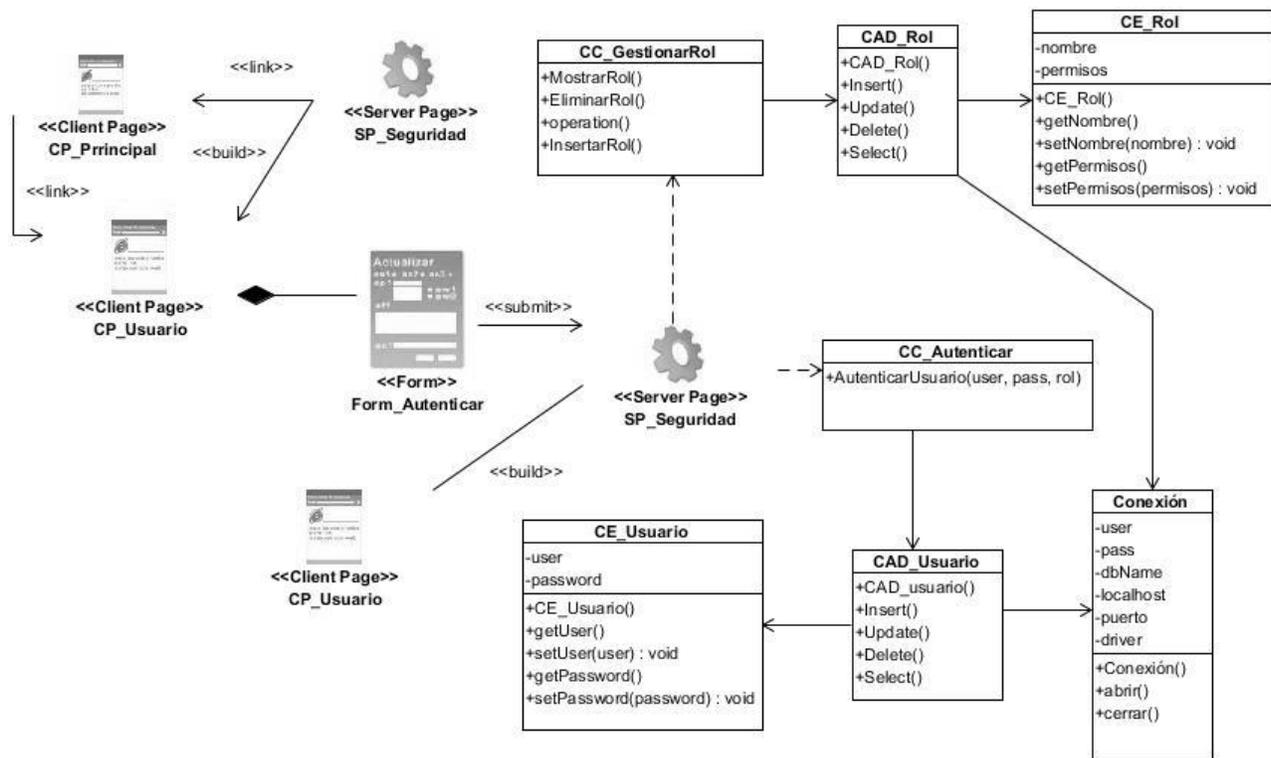


Figura 3 Diagrama del diseño de la HU autentificar usuario (elaboración propia)

2.4.2 Patrones de diseño

“Un patrón se define como una solución probada con éxito que aparece una y otra vez ante determinado tipo de problema en un contexto dado. Los patrones se definen por un nombre, un problema, una solución y las consecuencias de su aplicación. Este define una posible solución correcta para un problema de diseño dentro de un contexto dado, describiendo las cualidades invariantes de todas las soluciones” (Abreu, 2013).

Patrones GOF

Estos patrones están definidos por *The Gang of Four* (GOF) en procesos de desarrollo de *software* orientados a la web. El grupo GOF se dedica al análisis de los problemas recurrentes en el desarrollo de software y realizan una clasificación y agrupación a partir de dos criterios, su propósito y alcance (García, 2017).

Patrones Generales de Software para Asignación de Responsabilidades (GRASP)

GRASP son patrones generales de software para asignación de responsabilidades, es el acrónimo de "GRASP (*object-oriented design General Responsibility Assignment Software Patterns*)", se utiliza en el diseño orientado a objetos. Se considera que más que patrones propiamente dichos,

CAPÍTULO 2. Caracterización del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software (Departamento de Lenguajes y Sistemas Informáticos, 2017).

Drupal utiliza los patrones de diseño como: sencillo o instancia única (*singleton*), decorador (*decorator*), observador (*observer*), puente (*bridge*), cadena de responsabilidad (*chain of responsibility*) y comando (*command*). Estos permiten diseñar sistemas seguros y que a su vez cumplan con los estándares de diseño establecidos por normas internacionales para el desarrollo de aplicaciones Web (Drupal, 2015). Los patrones que serán utilizados para el desarrollo de la propuesta de solución son:

Instancia única (*Singleton*): Este patrón está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que cada clase tenga una instancia única y un punto de acceso a la misma. Se hace uso de este en la clase "Conexión", la cual es un objeto que utiliza una base de datos que es una instancia única para el proyecto en uso (Suárez y otros, 2013).

Observador (*observer*): Los módulos que implementan un hook determinado por evento de inserción o actualización de una determinada entidad, son declarados como observadores de dichas entidades con las que interactúan. En la propuesta de solución se utiliza el hook `_node_insert()`, que se ejecuta al insertar un tipo de contenido en específico, por lo que el módulo es declarado como observador ya que se encuentra a la espera de un evento de inserción. Se utiliza además en la implementación del hook `_node_update`.

Puente (*bridge*): La capa de abstracción de base de datos de Drupal se implementa de una forma similar al patrón bridge, los módulos necesitan ser escritos en una manera que es independiente del sistema de base de datos que se utiliza, en el caso de la propuesta de solución la implementación de cada uno de los hook se realiza independiente y estándar del gestor de bases de datos utilizado. Ejemplos de estos hook se encuentran el hook `_menu`, `_permission`, `_form_alter_id`, `_node_insert` y `_node_update` que realizan consultas a la base de datos sin importar de que tipo sea esta.

Cadena de responsabilidad (*chain of responsibility*): El sistema de menús de Drupal es la evidencia del patrón Chain of responsibility, en dependencia de la petición recibida por un módulo, se delega la responsabilidad a la función encargada de atenderla. De esta forma se continúa la cadena hasta que un módulo atienda la petición o la cadena se agote. La implementación del hook `_menu` es un ejemplo claro que genera un menú de configuración del módulo.

Comando (*command*): Muchos de los hook implementados en la propuesta de solución utilizan el patrón de command para reducir el número de funciones que son necesarias para funcionar, pasando la operación como un parámetro, junto con los argumentos. De hecho, el propio sistema de hook utiliza este patrón, por lo que los módulos no tienen que definir cada hook, sino más bien sólo los que se necesiten implementar.

2.5 Diagrama de despliegue

El diagrama de despliegue, representa de forma visual las relaciones físicas que existen entre los componentes de software y hardware en el sistema. El módulo se encuentra hospedado en un servidor Web (Apache 2.4) y se comunica con un SGBD. A continuación, se muestra el diagrama de despliegue propuesto para el sistema donde quedan definidas las relaciones entre los diferentes nodos físicos que lo componen.

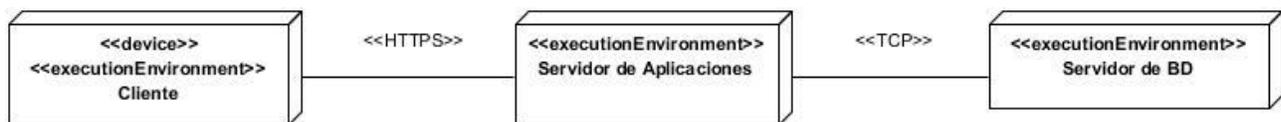


Figura 4 Diagrama de despliegue (elaboración propia).

A continuación, se describen los componentes del diagrama de despliegue anterior:

PC Cliente: La estación de trabajo cliente necesita un navegador web para conectarse al sistema hospedado en el servidor de aplicaciones utilizando el protocolo de comunicación HTTP/HTTPS.

Servidor de aplicación: Es la estación de trabajo que hospeda el código fuente de la aplicación y que les brinda a los usuarios las interfaces para realizar los procesos definidos por cada uno de los roles del sistema. Esta estación se comunica con el servidor de BD donde se almacenan los datos de la aplicación realizando la comunicación mediante el protocolo TCP/IP.

Servidor de BD: Este servidor es el encargado del almacenamiento de los datos del sistema. Se comunica con el servidor de aplicaciones del sistema, posibilitando el acceso mediante el usuario con privilegios para las operaciones determinadas a realizarse en el mismo.

2.6 Diagrama de datos

La permanencia de la información es una de las exigencias clave en el diseño de la arquitectura de una aplicación informática. Se considera que es de vital importancia la realización de un diseño de base de datos que cumpla con las buenas prácticas en la representación de las clases entidades y

CAPÍTULO 2. Caracterización del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

sus relaciones. Estas permitirán almacenar la información con un mínimo de redundancia, pero que a la vez faciliten su recuperación (García, 2017).

El sistema se propone hacer uso de una base de datos, puesto que necesita la información referente a las fichas técnicas y bases de cálculos. Como ya se ha definido anteriormente la información será almacenada con MySQL. Para dicha tarea se describen las colecciones de la base de datos:

CE_Ficha_Técnica: almacena la información de la ficha técnica creada a partir de la base de cálculo correspondiente.

CE_Base_Cálculo: almacena la información de la base de cálculo para crear posteriormente la ficha técnica.

CE_Usuario: almacena la información del usuario registrado con su respectivo perfil.

CE_Rol: almacena la información referente a los roles y sus respectivos permisos.

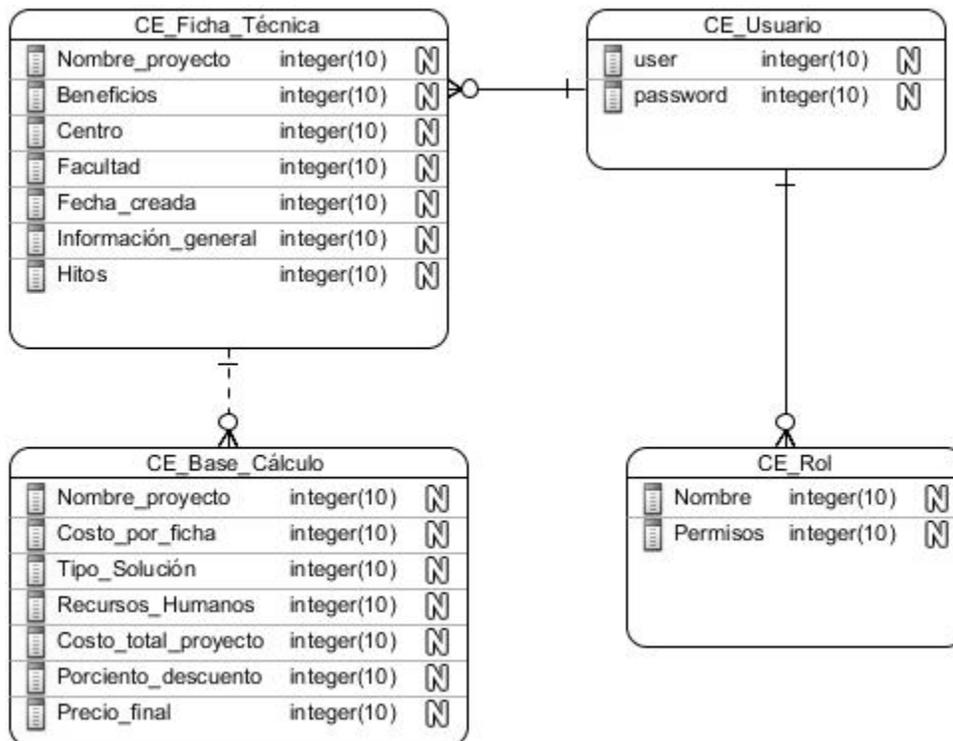


Figura 5 Diagrama de Modelo de Datos (elaboración propia)

2.7 Conclusiones parciales

En este capítulo se realizó el análisis, el diseño y la arquitectura en general del sistema de gestión de ofertas comerciales. Para cumplir dicha tarea se representaron y describieron los artefactos definidos por la metodología de manera que se entendiera el flujo de trabajo presente. Se representó el entorno en el que se ubica el problema a través del modelo de dominio, que constituye un punto de partida para el diseño del sistema. Los requerimientos funcionales y no funcionales obtenidos a partir del proceso de identificación de los requisitos, sirvieron de guía para desarrollar las distintas funcionalidades y de este modo satisfacer las necesidades detectadas. Los artefactos generados según la metodología de desarrollo utilizada y los patrones de arquitectura y diseño descritos, constituyeron una guía fundamental para la construcción de la propuesta de solución.

CAPÍTULO 3 “Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1”

En este capítulo se realiza la fase implementación y posteriormente las pruebas al sistema de gestión. Se describen los estándares de codificación empleados durante la implementación. Junto al proceso de implementación, el sistema debe ser expuesto a diferentes pruebas para darle validez a los requisitos funcionales y garantizar el óptimo funcionamiento del sistema.

3.1 Modelo de implementación

Según (Pressman, 2015) el modelo de implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se encuentran datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos.

3.1.1 Diagrama de componentes

Los diagramas de componentes son utilizados para estructurar el modelo de la implementación. Permiten modelar una vista estática del sistema, muestran la organización y las dependencias lógicas entre un conjunto de componentes del *software*, que pueden ser librerías, binarios, ejecutables y códigos fuentes.

A continuación, se muestran las descripciones de los componentes y el diagrama de componentes, este último teniendo a los requisitos del sistema a implementar.

CAPÍTULO 3: Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

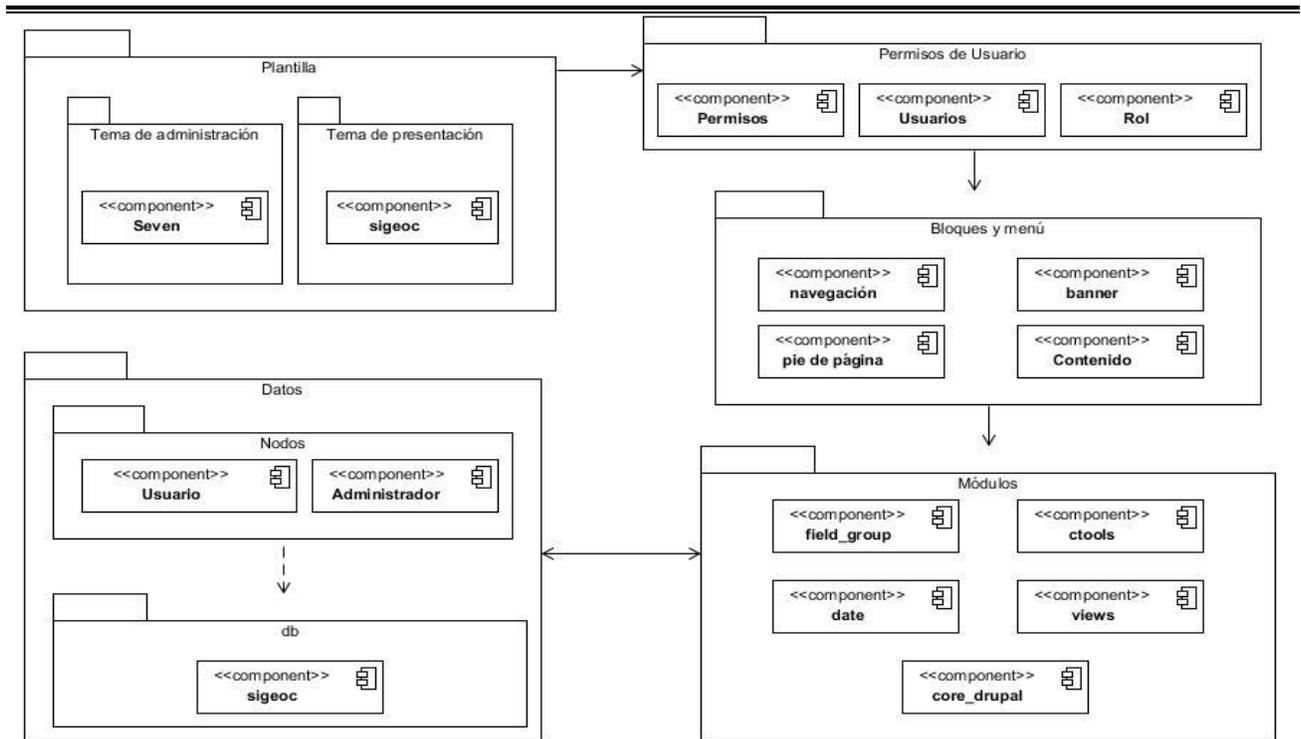


Figura 6 Diagrama de componentes (elaboración propia)

A continuación, se describen cada uno de los componentes del diagrama:

index.php: es el punto de inicio de la aplicación, a partir de esta entrada se solicitan los diferentes módulos presentes en el sistema gestor de contenido Drupal 7.

themes: conjunto de temas que representan la forma en que se visualiza el sistema, los mismos vienen por defecto en el CMS Drupal 7.

profiles: contiene los perfiles de instalación del CMS Drupal.

misc: este directorio incluye un grupo de archivos necesarios para el sistema como: *JavaScript*, *CSS* e imágenes.

sites: se encuentran elementos extras y modificaciones que se añaden al sistema general.

scripts: contienen utilidades adicionales que no utiliza Drupal directamente, pero que pueden ser utilizadas desde la línea de comando *Shell*.

includes: contiene los ficheros indispensables para el funcionamiento de Drupal.

modules: conjunto de módulos que son parte del núcleo de Drupal y permiten el funcionamiento del sistema.

3.2 Estándares de codificación utilizados

Los estándares de código resultan importantes en cualquier proyecto de desarrollo los mismos ayudan a asegurar que el código tenga una alta calidad, menos errores, pueda ser mantenido fácilmente y sea legible. Para el desarrollo del módulo se utilizan los estándares de codificación definidos por Drupal en (Coding_Standards, 2016). A continuación, se explican y ejemplifican los estándares de codificación utilizados en la propuesta de solución.

Operadores

Todos los operadores binarios como: +, -, !, =, ==, >, deben tener un espacio antes y después del operador, los operadores unarios como ++, no utilizan espacios entre el operador y la variable.

```
$elements = drupal_get_form('update_script_selection_form');  
$output = drupal_render($elements);
```

Figura 7 Ejemplo de uso de los operadores en el archivo update.php

Estructuras de control

Las estructuras de control incluyen *if*, *for*, *while*, *switch*, entre otros. Las mismas deben presentar un espacio entre la palabra de control (*if*, *for*) y el paréntesis que abre, para distinguirlos de las llamadas a las funciones. La llave de apertura ({} se situará en la misma línea que la definición de la estructura de control, separada por un espacio como se demuestra en la figura 7.

```
if (!isset($update['start'])) {  
    $form['start'][$module] = array(  
        '#type' => 'item',  
        '#title' => $module . ' module',  
        '#markup' => $update['warning'],  
        '#prefix' => '<div class="messages warning">',  
        '#suffix' => '</div>',  
    );  
    $incompatible_updates_exist = TRUE;  
    continue;  
}
```

Figura 8 Ejemplo de uso de la estructura de control if en el archivo update.php

CAPÍTULO 3: Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

La estructura switch se utilizará de la siguiente manera:

```
switch ($op) {
  // update.php ops.

  case 'selection':
    if (isset($_GET['token']) && drupal_valid_token($_GET['token'], 'update')) {
      $output = update_selection_page();
      break;
    }

  case 'Apply pending updates':
    if (isset($_GET['token']) && drupal_valid_token($_GET['token'], 'update')) {
      // Generate absolute URLs for the batch processing (using $base_root),
      // since the batch API will pass them to url() which does not handle
      // update.php correctly by default.
      $batch_url = $base_root . drupal_current_script_url();
      $redirect_url = $base_root . drupal_current_script_url(array('op' => 'results'));
      update_batch($_POST['start'], $redirect_url, $batch_url);
      break;
    }

  case 'info':
    $output = update_info_page();
    break;

  case 'results':
    $output = update_results_page();
    break;

  // Regular batch ops : defer to batch processing API.
  default:
    update_task_list('run');
    $output = _batch_page();
    break;
}
```

Figura 9 Ejemplo de uso de la estructura de control switch en el archivo update.php

Arreglos (arrays)

Los valores dentro de un array (o matriz) se deben separar por un espacio (después de la coma que los separa). El operador => debe separarse por un espacio a ambos lados.

```
$form['start'][$module] = array(
  '#type' => 'item',
  '#title' => $module . ' module',
  '#markup' => $update['warning'],
  '#prefix' => '<div class="messages warning">',
  '#suffix' => '</div>',
);
```

Figura 10 Ejemplo de declaración de un array en el archivo update.php

Uso de comillas

Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres. Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto o cuando el texto puede incluir alguna comilla simple.

```
drupal_set_title('Drupal database update');
```

Figura 11 Ejemplo de uso de las comillas simples en el archivo update.php

```
$messages[] = '<li class="success">' . $query['query'] . '</li>';
```

Figura 12 Ejemplo de uso de las comillas dobles en el archivo update.php

Utilización del punto y coma (;) en código PHP

En Drupal es siempre obligatorio el terminador de línea (;).

```
$skip_warnings = !empty($_GET['continue']);  
update_check_requirements($skip_warnings);  
  
$op = isset($_REQUEST['op']) ? $_REQUEST['op'] : '';
```

Figura 13 Ejemplo de uso del punto y coma en el archivo update.php

Nombres de funciones

Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por un guión bajo. Se debe incluir siempre como prefijo el nombre del módulo. En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior.

```
function update_selection_page() {  
  drupal_set_title('Drupal database update');  
  $elements = drupal_get_form('update_script_selection_form');  
  $output = drupal_render($elements);  
  
  update_task_list('select');  
  
  return $output;  
}
```

Figura 14 Ejemplo de nombres de las funciones en el archivo update.php

Nombres de variables

Las variables deben ser nombradas en minúsculas y las palabras deben estar separadas por un guión bajo.

```
$updates = update_get_update_list();  
$starting_updates = array();
```

Figura 15 Ejemplo de nombres de variables en el archivo update.php

3.4 Plan de iteraciones

El plan de entrega del producto se divide en dos iteraciones, teniendo como base la prioridad y complejidad de cada uno de los requisitos del sistema, partiendo de la definición de las historias de usuarios. Al concluir la última iteración, la aplicación se encuentra lista para su despliegue. A continuación, se detallan los elementos analizados en cada iteración.

Iteración 1

En la primera iteración se implementan las historias de usuarios con mayor prioridad, obteniendo al final de la misma una primera versión de prueba en el proceso de desarrollo incremental y dotando al sistema de sus primeras funcionalidades.

Iteración 2

En la segunda iteración se realiza la implementación de los requisitos correspondientes a las historias de usuarios con prioridad alta para el cliente, además se corrigen los errores encontrados en las historias de usuarios desarrolladas durante la primera iteración y en la segunda. Al concluir se obtiene la versión 1.0 del producto final.

Duración de las iteraciones

Para lograr una correcta organización en el desarrollo de la aplicación, se hace necesaria la creación de un plan de duración de iteraciones. Este plan se confecciona con el objetivo de establecer un orden en el desarrollo de los requisitos de la aplicación, así como el tiempo destinado a cada una de las iteraciones (Tabla 3).

Iteraciones	Descripción de la iteración	Orden de las HU a implementar	Duración total
1	Desarrollo de las	1,2,3,4,5,7,8,9,10,11,12,13,14	5 semanas

CAPÍTULO 3: Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

	historias de usuario de prioridad muy alta		
2	Desarrollo de las historias de usuario de prioridad media	6,15,16	1 semana

Tabla 3 Tiempo de duración de las iteraciones (elaboración propia).

3.4 Pruebas de software

Las pruebas de software son un conjunto de herramientas, técnicas y métodos que evalúan la excelencia, el desempeño de un software, involucra las operaciones del sistema bajo condiciones controladas y evalúa los resultados. Las técnicas para encontrar problemas en un programa son variadas y van desde el uso del ingenio por parte del personal de prueba hasta herramientas automatizadas que ayudan a aliviar el peso y el costo de tiempo de esta actividad (Pressman, 2016).

Una vez que se implementa el código del sistema es necesario probarlo para encontrar y enmendar la mayor cantidad de errores antes de entregarlo al cliente. El objetivo de este proceso es diseñar casos de pruebas que tengan una alta probabilidad de encontrar errores. Para alcanzar este objetivo existen técnicas de pruebas de software, que contienen directrices sistemáticas para comprobar la lógica interna y de interfaces de los componentes del sistema, así como los dominios de entrada y salida para identificar errores funcionales y de desempeño.

A continuación, se describen los tipos de pruebas de software aplicadas, así como los métodos y técnicas empleadas para la evaluación de la propuesta de solución.

3.5 Aplicación de las pruebas de software

Al terminar la implementación del sistema es necesario realizarle pruebas con el objetivo de detectar errores en la aplicación y la documentación. Este es un proceso de mucha importancia pues proporciona una medida de calidad del mismo, siempre llevándolo a cabo de una forma correcta.

A continuación, se presentan los tipos de pruebas de software aplicados al sistema implementado. El objetivo principal de las mismas es la detección de las no conformidades respecto a los requisitos

funcionales del sistema, el rendimiento del sistema, la aceptación del usuario y la correcta integración entre los componentes.

3.5.1 Pruebas Funcionales

Las pruebas funcionales son aquellas que se llevan a cabo sobre la interfaz del software sin prestar atención al código, por lo que los casos de prueba son creados con el objetivo de demostrar que la entrada es aceptada de forma adecuada y que se produce una salida correcta. El diseño de esta prueba se realiza con la intención de detectar funciones incorrectas o ausentes, errores en accesos a bases de datos externas, errores de interfaz, errores de rendimiento, y errores de inicialización y de terminación (Pressman, 2016).

Prueba de caja negra

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa. Los casos de prueba de la caja negra pretenden demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma correcta.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Técnica: Particiones de equivalencia

- Divide el dominio de entrada de un programa en clases de datos (clases equivalentes) de los que se pueden derivar casos de prueba.
- Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada (valor específico, rango de valores, condición lógica, etc.).

Pasos:

- 1) Identificar todas las clases de equivalencia.
- 2) Definir los casos de prueba:
 - Incluir en un caso tantas clases válidas como sea posible, hasta que todas estén cubiertas.
 - En casos aparte, las clases inválidas.
- 3) Definir valores de datos para las pruebas

CAPÍTULO 3: Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

Condiciones de ejecución

Autenticar Usuario

Usuario: puede contener solo letras, números (no puede contener caracteres extraños) y una cantidad de hasta 15 caracteres.

Contraseña: valor alfanumérico y puede contener caracteres extraños y letras mayúsculas y minúsculas.

Campo	Clases Válidas	Clases Inválidas	Juegos de Datos
usuario	Caracter<= 15	Caracter > 15	Cajanegra1234567
	Caracter= números y letras	Carácter != números y letras	*cajanegra1234*
Contraseña	Caracter= números y letras	Carácter != números y letras	Cajanegra?..
	Caracter<= 20	Caracter > 20	Cajanegrayblanca123***
	Caracter= @, *	Caracter != @, *	Cajanegra?***
	Caracter = mayúsculas y minúsculas	Caracter != mayúsculas y minúsculas	Pruebacajanegra

Tabla 4 Resultados de aplicar la prueba de caja negra (elaboración propia).

Prueba de caja blanca

Las pruebas de caja blanca (también conocidas como pruebas de caja de cristal o pruebas estructurales) se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. El ingeniero de pruebas escoge distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y cerciorarse de que se devuelven los valores de salida adecuados.

Técnica “Camino básico”:

Pasos:

1. Dibujar el grafo de flujo asociado de la unidad.
2. Calcular la complejidad ciclomática del grafo (V(G)).

CAPÍTULO 3: Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

3. Determinar un conjunto básico de caminos independientes.

4. Preparar al menos 1 caso de prueba por cada camino.

Método: Caja Blanca `camino básico`

```
function administracion_preprocess_html(&$variables) {  
    global $user; //1  
    // Agregar css propios a pagina de administración  
    if(!array_search("administrator",$user->roles)){ //2  
        $theme_path = drupal_get_path('theme', 'seven'); //3  
        drupal_add_css(drupal_get_path('module', 'administracion') . '/administracion_custom.css',  
array('group' => CSS_THEME)); //4  
        drupal_add_js(drupal_get_path('module', 'administracion').'/administracion.js'); //5  
    }  
} //6
```

Grafo resultante:

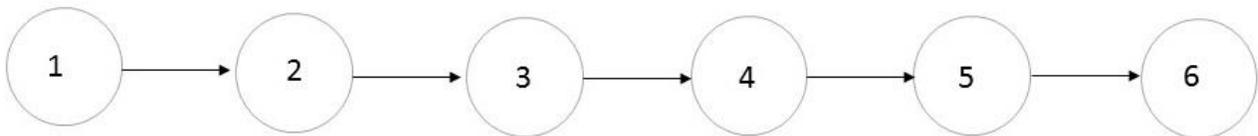


Figura 16 Grafo resultante de del método caja blanca camino básico (elaboración propia)

Complejidad Ciclomática:

$$V(G) = \# \text{ regiones} = 1$$

$$V(G) = A - N + 2 = 5 - 6 + 2 = 1$$

$$V(G) = P + 1 = 1$$

Caminos lógicos:

1-2-3-4-5-6

CAPÍTULO 3: Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

Resultados de las pruebas por caso de pruebas

Identificador del Caso de Prueba	Caso de Prueba	Entradas	Resultado Esperado	Resultado de la Prueba
Crear usuario	Gestionar usuario	SusanaCalasjerez	El sistema debe validar el campo	El sistema permite la creación de un usuario.
Crear usuario	Gestionar usuario	Sus@n@Ca1asJerez	El sistema no permite la creación del usuario	El sistema no permite crear un usuario con un nombre no válido.
Crear usuario	Gestionar usuario	Micontraseñaparaelsistema	El sistema debe validar el campo	El sistema permite la creación de un usuario.
Crear usuario	Gestionar usuario	123	El sistema no permite la creación del usuario	El sistema no permite crear un usuario con una contraseña que no cumple con los requerimientos.
Crear usuario	Gestionar usuario	scalas@estudiantes.uci.cu	El sistema debe validar el campo	El sistema permite la creación de un usuario.
Crear usuario	Gestionar usuario	scalas@gmail.com	El sistema no permite la creación del usuario	El sistema no permite crear un usuario con un dominio que no sea

CAPÍTULO 3: Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

				perteneciente al uci.cu
--	--	--	--	----------------------------

Tabla 5 Resultados de las pruebas por caso de pruebas (elaboración propia).

3.5.2 Pruebas de seguridad

Las pruebas de seguridad garantizan que los usuarios estén restringidos a funciones específicas o que su acceso esté limitado únicamente a los datos que están autorizados a acceder. Sólo aquellos usuarios autorizados a acceder al sistema son capaces de ejecutar las funcionalidades disponibles. El objetivo fundamental de este tipo de pruebas es comprobar los niveles de seguridad lógica del sistema.

Resultados de las pruebas de seguridad

Para evaluar la seguridad de la aplicación se hizo uso de la herramienta Acunetix WVS v8 el cual devolvió los siguientes resultados.

Categorías de vulnerabilidades	Cantidad de Errores
Formularios HTML sin protección CSRF	6
Credenciales de usuarios enviadas en texto plano	12
Campos de contraseña con auto completamiento activado	0
Total	18

Tabla 6 Resultados de la aplicación de la herramienta Acunetix WVS v8 para pruebas de seguridad (elaboración propia).

Se corrigen los errores relacionados con las funciones del sistema y los restantes errores corresponden al sistema Drupal y no constituyen problemas de seguridad del mismo.

3.5.3 Pruebas de rendimiento

Mediante las pruebas de rendimiento es posible hallar tendencias y comportamientos para los elementos de una aplicación, los cuales generan bajo rendimiento. Este tipo de pruebas permiten identificar cuellos de botella, capacidad de concurrencia de usuarios, tiempos de respuesta de operaciones de negocio a nivel de sistema, establecer un marco de referencia para pruebas futuras,

CAPÍTULO 3: Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

determinar el cumplimiento de los objetivos de rendimiento y requisitos no funcionales, entre otros (V&v_Quality_S.A, 2015).

Pruebas de estrés: Mediante las pruebas de estrés es posible identificar la capacidad de respuesta de un sistema bajo condiciones de carga extrema, representadas por una alta concurrencia de Usuarios y/o procesos, una vez realizadas las pruebas de estrés se podrá conocer el punto de quiebre del aplicativo en términos de capacidad de respuesta, con lo cual será posible establecer acciones de optimización en diferentes niveles para asegurar una mejor capacidad de concurrencia de usuarios y/o procesos que se verá reflejada en una óptima operación de negocio (V&v_Quality_S.A, 2015).

Pruebas de carga: Mediante la ejecución de las pruebas de Carga es posible identificar la capacidad de recuperación de un sistema cuando es sometido a cargas variables tanto de usuarios como de procesos. Al realizar las pruebas de carga se puede determinar el tiempo de respuesta de todas las transacciones críticas del sistema y encontrar cuellos de botella de la misma (V&v_Quality_S.A, 2015).

Resultados de las pruebas de rendimiento

Para las pruebas de rendimiento se utiliza el software Apache Jmeter. Para ello se definen las propiedades de las PC utilizadas tanto la cliente como la utilizada como servidor.

Hardware de prueba (PC cliente):

- Tipo de procesador: Intel(R) Pentium(R) 3558U @ 1.70GHz x2.
- RAM: 4 GB.
- Tipo de Red: Ethernet 10/100Mbps.

Hardware de prueba (PC servidor):

- Tipo de procesador: Intel(R) Pentium(R) 3558U @ 1.70GHz x2.56
- RAM: 4 GB.
- Tipo de Red: Ethernet 10/100Mbps.

Software instalado en ambas PC:

- Tipo de servidor web: Apache 2.4.

CAPÍTULO 3: Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

- Plataforma: SO Linux Mint 17.3.
- Servidor de BD: MySQL v5.7.

3.5.4 Pruebas de aceptación

Esta es la etapa final en el proceso de pruebas, antes de que el sistema sea aceptado para uso operacional. El sistema se pone a prueba con datos suministrados por el cliente del sistema, en vez de datos de prueba simulados. Las pruebas de aceptación revelan los errores y las omisiones en la definición de requerimientos del sistema, ya que los datos reales ejercitan el sistema en diferentes formas a partir de los datos de prueba (Sommerville , 2015).

Caso de prueba de aceptación
Nombre de la historia de usuario: Crear ficha técnica
Nombre de la persona que realiza la prueba: Adyana Pileta Gammalame
Descripción de la prueba: El sistema permite crear las fichas técnicas.
Condiciones de ejecución: El usuario debe autenticarse en el sistema para poder acceder a la opción Crear Ficha Técnica.
Entrada/Pasos de ejecución: <ol style="list-style-type: none">1. Acceder al sistema2. Autenticarse en el sistema3. Crear ficha técnica
Resultado esperado: El sistema crea la ficha técnica, esta queda guardada en el sistema y es posible reutilizarla como plantilla para realizar otras.
Evaluación de la prueba: Satisfactoria

Tabla 7 Caso de prueba de aceptación para la historia de usuario "Crear ficha técnica"(elaboración propia)

Al terminar las pruebas de aceptación se obtuvo un total de 5 no conformidades en la primera iteración, de las cuales 3 fueron resueltas y 2 no procedían. En la segunda iteración se obtuvieron 4 no conformidades donde todas fueron resueltas y se realizó una tercera iteración para comprobar los datos obtenidos en la segunda iteración y no arrojó no conformidades . Durante las 3

CAPÍTULO 3: Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

iteraciones no quedó ninguna no conformidad por resolver. Los resultados correspondientes a estas pruebas se muestran en la figura 16.

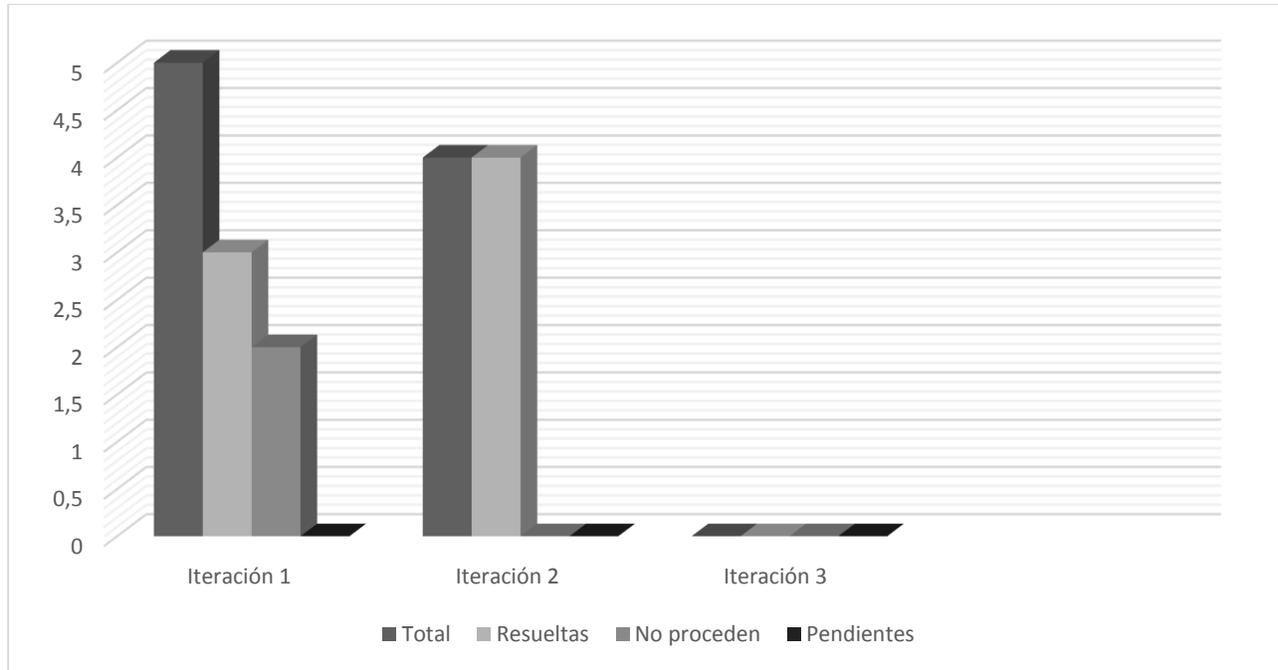


Figura 17 Resultados de las pruebas de aceptación (elaboración propia)

3.6 Validación de la Hipótesis

Para la validación de la hipótesis de investigación se utiliza el método criterio de expertos en su variante Delphi (Sánchez, 2015) empleando los siguientes pasos:

- Identificación de los posibles expertos.
- Selección de los expertos.
- Realización de la consulta a los expertos y procesamiento y valoración de la información obtenida.

Para identificar los posibles expertos se tuvieron en cuenta, la experiencia profesional en relación con el proceso de confección de la oferta y el desarrollo en Drupal 7. En la siguiente tabla se muestran los expertos seleccionados.

CAPÍTULO 3: Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

	Experto	Entidad	Años de experiencia
1	Adyana Pileta Gammalame	CIDI	1
2	Kilmer Hernández Ávila	CISED	1
3	Yojahny Chávez Marrero	CIDI	3

Tabla 8 Expertos seleccionados para realizar la validación de la hipótesis (elaboración propia)

Luego de seleccionados los expertos, se sometió a su consideración un instrumento para la validación del sistema de gestión de ofertas comerciales. El instrumento se compone de 5 sentencias, así como 5 categorías evaluativas que permitan conocer la opinión de los expertos. Las categorías evaluativas empleadas fueron: muy adecuado (MA), bastante adecuado (BA), adecuado (A), poco adecuado (PA) e inadecuado (I). Anexo 1

Se calcula el coeficiente de Kendall que permite analizar la concordancia en las valoraciones realizadas por los expertos (Sampieri y Fernández, 2010). En este caso el coeficiente de concordancia (W) será un índice de la divergencia del acuerdo efectivo entre los expertos. El coeficiente de concordancia de Kendall se obtiene de la expresión

$$W=12S/K2(N3-N) \quad (4) \text{ Coeficiente de Kendall}$$

Donde S representa el cuadrado de las desviaciones medias, K el número de expertos y N el número total de aspectos a evaluar. El valor de W oscila entre 0 y 1. El valor de 1 significa una concordancia de acuerdos total y el valor de 0 un desacuerdo total. Se aplica además la Prueba de Significación de Hipótesis para comprobar el grado de significación de Kendall, planteándose la hipótesis nula y la alternativa de la siguiente forma: donde H0: no existe concordancia entre los expertos y H1: existe concordancia entre los expertos.

$$X2=K(N-1) W \quad (5) \text{ Chi Cuadrado}$$

$$X2=0.352$$

El X2 calculado se compara con el tabulado en la tabla del percentil de la distribución X2. Para tener un 95% de confianza se utilizará $\alpha=0.05$. Si se cumple que X2 calculada < X2(α , N-1), se obtiene que 0.352<9,4877 entonces es válida la hipótesis alternativa H1 de que existe concordancia entre

CAPÍTULO 3: Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

los expertos. Los criterios aportados por los expertos se someten a una prueba estadística no paramétrica que permite concluir qué valoración final tiene cada uno de los aspectos a evaluar. Para los datos anteriores se debe confeccionar una distribución de frecuencia a partir de los datos primarios para cada uno de los aspectos sometidos a consulta (Castro, 2016).

Categorías Evaluativas	Frecuencia Absoluta	Frecuencia Relativa
Muy Adecuado	23	0,92
Bastante Adecuado	1	0,04
Adecuado	1	0,04
Poco Adecuado	0	0
Inadecuado	0	0

Tabla 9 Distribución de frecuencias a partir de datos primarios (elaboración propia)

Los resultados obtenidos mediante la validación se pueden observar mediante la siguiente figura.

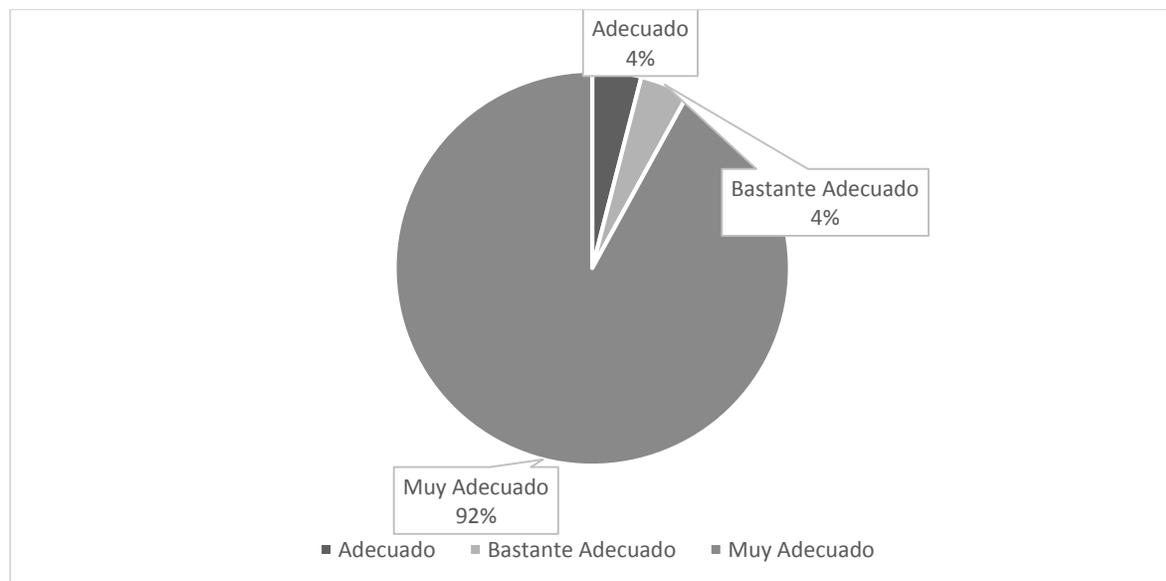


Figura 18 Resultados obtenidos de la validación de hipótesis (elaboración propia)

CAPÍTULO 3: Implementación y pruebas del sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1

De acuerdo con los datos el 92% de los aspectos analizados fueron valorados de muy adecuado, el 4% de bastante adecuado y un 4% de adecuado. El análisis de los resultados obtenidos de la consulta de expertos permitió identificar que existe una coincidencia en las valoraciones realizadas sobre el alto valor que tiene la utilización del sistema de gestión de ofertas comerciales. Los indicadores fueron evaluados satisfactoriamente, evidenciando la calidad de la propuesta presentada.

3.7 Interfaz principal

Una vez finalizado el desarrollo del software es posible visualizar la pantalla principal del Sistema de gestión de ofertas comerciales para proyectos productivos en la Facultad 1, donde se observa el resultado obtenido durante la implementación de las historias de usuario descritas en el capítulo anterior.



Figura 19 Interfaz del sistema

3.8 Conclusiones parciales

La confección del diagrama de componentes permitió observar la integración de los componentes de software. Las pruebas realizadas al sistema contribuyeron al impacto positivo del producto. La aplicación de las pruebas realizadas al sistema y la consulta a expertos permitió identificar las principales deficiencias en el desarrollo del sistema, así como la estrategia para solucionar los errores detectados y obtener un producto con un alto valor, pertinencia y utilidad para la gestión de ofertas comerciales.

CONCLUSIONES GENERALES

El desarrollo de la presente investigación ha posibilitado el cumplimiento de los objetivos y tareas propuestas, por lo que se arriban a las siguientes conclusiones:

- El estudio de los referentes teóricos y el análisis de las diferentes herramientas para la gestión de ofertas comerciales permitieron determinar las características que constituyen la base para el diseño de las funcionalidades que se definen en la propuesta de solución.
- El sistema cuenta con una interfaz agradable para el usuario, y de fácil utilización, una página principal que presenta un menú de navegación que muestra las opciones siguientes: crear ficha técnica, mostrar fichas técnicas, crear base cálculo y mostrar base de cálculo, así como otras páginas secundarias.
- Las técnicas de validación aplicadas al sistema, demostraron que el mismo es una solución funcional, segura, con un rendimiento adecuado.
- La validación de la hipótesis con criterio de expertos demostró que el sistema contribuye a la disminución del tiempo en la realización de la oferta.

RECOMENDACIONES

Una vez concluida la investigación y el desarrollo de la propuesta de solución, la autora de la presente recomienda:

- Implementar nuevas funcionalidades dentro del sistema que permitan seguir agilizando el trabajo con la información referente a las ofertas comerciales en la Facultad 1 y utilizar un nuevo módulo que permita exportar la base de cálculo correspondiente a la ficha técnica generada.
- Integrar la propuesta de solución con el sistema de gestión de proyectos (GESPRO).
- Sirva como base para investigaciones futuras.

Bibliografía

Ameijide, Laura. 2016. *Gestión de proyectos según el PMI*. Catalunya, España : s.n., 2016.

Apache Jmeter. 2015. 2015.

Benítez, Javier. 2011. *Enfoque más Comercial de la Fase de Definición de un Proyecto Informático*. 2011.

Byron y Berry. 2012. 2012.

Cabeza Chávez, Yanet. 2015. *Módulo de JAVA para la herramienta Auditoría de Código Fuente*. La Habana : s.n., 2015. Trabajos de diplomas.

Calidad y software. [En línea] [Citado el: 10 de marzo de 2019.]
http://www.calidadyssoftware.com/testing/pruebas_unitarias1.php..

Centro de Ideoinformática_ Universidad de las Ciencias Informáticas. UCI. 2018. 2018.

Centros de Desarrollo_ Universidad de las Ciencias Informáticas. UCI. 2018. 2018.

Chávez, Yojahny. 2016. *Trabajo de diploma :Módulo de publicación automática de contenidos web en redes sociales*. La Habana : s.n., 2016.

Coding_Standards. 2016. *Coding Standards*. 2016.

Drupal 7. 2018.

Ecured. *Ecured*. [En línea] [Citado el: 14 de enero de 2019.] http://www.ecured.cu/Patrones_GoF.

García, Alexander. 2017. *Componente para el cálculo de la relevancia de información en el buscador Orión*. 2017.

Garzón. 2014. 2014.

Hernández, Enrique Orallo. 2015. *El Lenguaje Unificado de Modelado (UML)* . [En línea] 2015. [Citado el: 6 de 12 de 2017.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.

Historia_ Universidad de las Ciencias Informáticas. UCI. 2018. 2018.

Informáticas, Universidad de las Ciencias. 2018. 2018.

Kemppainen. 2015. 2015.

Kholodkov. 2015. 2015.

Larman , Craig. 2003. *Modelo de dominio*. [aut. libro] Craig Larman. *UML y Patrones*. Prentice Hall : s.n., 2003.

López Martínez , J. A. 2015. *Sistema para la gestión de Oportunidades de Negocio para el Centro de Informática Industrial*. 2015.

- Maida, Esteban Gabriel. 2015.** Metodologías de Desarrollo de Software . [En línea] diciembre de 2015. [Citado el: 5 de 12 de 2017.] <http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf>.
- 2018.** MDN web docs. *MDN web docs*. [En línea] 2018. [Citado el: 6 de enero de 2019.] <https://developer.mozilla.org/es/docs/Web/JavaScript>.
- Modelado de sistemas con UML. *Modelado de sistemas con UML*. [En línea] [Citado el: 14 de enero de 2019.] <https://www.ibiblio.org/pub/linux/docs/LuCaS/Tutoriales?doc-modelado-sistemas-UML/multiple-html/x219.html>.
- Navaja Ojeda, Antonio. 2012.** Guía completa de CSS3. Guía completa de CSS3. [En línea] 2012. [Citado el: 20 de noviembre de 2018.] <https://openlibra.com/es/book/guia-completa-de-css3>.
- Nedelcu. 2014.** 2014.
- Nginx. 2015.** 2015.
- Pandiyán y Singhal. 2015.* 2015.
- Prácticas de software . *Prácticas de software* . [En línea] [Citado el: 14 de enero de 2019.] https://www.practicadesoftware.com.ar/2015/03/patrones_grasp.
- Pressman, Roger. 2015.** *Ingeniería de Software: Un Enfoque Práctico Quinta Edición*. Nueva York : Estados Unidos: McGraw-Hill, 2015.
- . **2005.** *Ingeniería de software: Un enfoque práctico Sexta Edición*. Nueva York. : Estados Unidos: McGraw-Hill / Interamericana de México, 2005.
- . **2016.** *Ingeniería del software, un enfoque práctico*. 2016.
- Pressman, Roger S. 2016.** Ingeniería del software.Un enfoque práctico. [En línea] 2016. [Citado el: 10 de diciembre de 2018.] <http://cotana.informatica.edu.bo/downloads/Id-Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF>.
- Rattinger, Alvaro. 2018.** Qué es mercadotecnia y por qué es importante. *Revista Merca2.0*. [En línea] 19 de 10 de 2018. [Citado el: 28 de 05 de 2019.] <https://www.merca20.com/que-es-mercadotecnia-definicion/>.
- Real Academia Española. 2018.* 2018.
- Rodríguez, Tamara Sánchez. 2015.** *Metodología de desarrollo para la Actividad productiva de la UCI*. 2015.
- RUMBAUCH, James, JACOBSON, Ivar y BOOCH, Grady.** *El Lenguaje Unificado de Modelado*. s.l. : Manual de Referencia.
- Sánchez, José, Vázquez, Guillermo y Mejía, Juan. 2017.** 65, Bogotá, Colombia : s.n., 2017, Vol. 27. ISBN 0121-5051.
- Sheltren, Newton y Catchpole. 2014.* 2014.
- Sommerville , Ian. 2015.** *Ingeniería de Software 9na s.l.* s.l. : Adison-Wesley, 2011, 2015. ISBN 978-0-13-703515-1.

REFERENCIAS BIBLIOGRÁFICAS

- Sommerville. 2005.** SlideShare. [En línea] 2005. [Citado el: 10 de diciembre de 2018.]
https://es.slideshare.net/jasc_584/ingenieriadesoftware-iansommerville7maedicion-9417118.
- Suárez, Daniel. 2016.** *Módulo para la gestión del catálogo anual de superación profesional y servicios.* 2016.
- Tarazona T., César H. 2016.** *Amenazas informáticas y seguridad de la información .* 2016.
- Tolmison. 2010.** 2010.
- Tsantalis. 2016.* 2016.
- Unified Modeling Language. 2015.* 2015.
- V&v_Quality_S.A. 2015.** *V&v Quality S.A.* 2015.
- Van Rossum , Guido. 2009.** Tutorial Python version 2.6.2. [En línea] 2009. [Citado el: 22 de noviembre de 2018.] <http://ralsina.me/static/tutorial-8.pdf>.
- Vasundhara, Shaikh. 2015.** *Shaikh y Vasundhara.* 2015.
- Verma, Bansal y Pandey. 2015.* 2015.
- Visual Paradigm. 2016.* 2016.
- W3C. 2018.** 2018.
- What is. 2015.* 2015.
- WordPress.** Ingeniería del Software . [En línea] [Citado el: 14 de enero de 2019.]
<https://arlethparedes.com/2012/08/27/patrones-de-arquitectura-vs-patrones-de-diseño/>.

Anexos

Anexo 1

Entrevista realizada al asesor de mercadotecnia de cada centro productivo de la Facultad 1.

Objetivo: Conocer, analizar las no conformidades con el proceso de gestión de ofertas comerciales en la universidad y determinar los requisitos que debe cumplir el sistema.

1. ¿Cómo se realiza el proceso?

__ manual __ automáticamente

2. ¿Es engorroso el proceso de confección de la oferta?

__ si __ no

3. ¿Cuántas ofertas pueden ser realizadas en un mismo mes?

__2 a 3 __ más de 3 (en este caso especificar cuantas__)

4. ¿Qué funcionalidades principales debería tener el sistema?

Anexo 2

Descripción de las historias de usuario

Tabla 10 Historia de usuario asignar rol

Historia de usuario	
Número: 2	Nombre Historia de Usuario: Asignar Rol
Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad : Alta	Tiempo Estimado: 1 día
Riesgo en Desarrollo: Medio	Tiempo Real: 2 días

Descripción: El sistema permitirá asignar los roles y permisos a cada usuario para la utilización del sistema.
Observaciones: Los usuarios administradores, y editores de contenido podrán utilizar el sistema para la gestión de ofertas comerciales. El sistema mostrará las opciones para escoger los roles de cada usuario.
Prototipo de interfaz: No aplica

Tabla 11 Historia de usuario crear roles

Historia de usuario	
Número: 3	Nombre Historia de Usuario: Crear rol
Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad : Alta	Tiempo Estimado: 1 día
Riesgo en Desarrollo: Medio	Tiempo Real: 2 días
Descripción: El sistema permitirá crear un rol que dependiendo de los permisos podrá utilizar el sistema.	
Observaciones: El sistema mostrará las opciones para crear los roles de cada usuario.	
Prototipo de interfaz: No aplica	

Tabla 12 : Historia de usuario mostrar rol

Historia de usuario	
Número: 4	Nombre Historia de Usuario: Mostrar rol
Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad : Alta	Tiempo Estimado: 1 día
Riesgo en Desarrollo: Medio	Tiempo Real: 2 días
Descripción: : El sistema permitirá mostrar el rol que anteriormente fue creado.	

Observaciones: El sistema mostrará los roles disponibles para cada usuario.
Prototipo de interfaz: No aplica

Tabla 13 Historia de usuario modificar rol

Historia de usuario	
Número: 5	Nombre Historia de Usuario: Modificar rol
Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad : Alta	Tiempo Estimado: 1 día
Riesgo en Desarrollo: Medio	Tiempo Real: 2 días
Descripción: El sistema permitirá modificar el rol que anteriormente fue creado.	
Observaciones: El sistema permitirá modificar cada uno de los roles creados.	
Prototipo de interfaz: No aplica	

Tabla 14 Historia de usuario eliminar rol

Historia de usuario	
Número: 6	Nombre Historia de Usuario: Eliminar rol
Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad : Media	Tiempo Estimado: 1 día
Riesgo en Desarrollo: Medio	Tiempo Real: 2 días
Descripción: El sistema permitirá eliminar el rol que anteriormente fue creado.	
Observaciones: El sistema permitirá eliminar cada uno de los roles creados.	
Prototipo de interfaz: No aplica	

Tabla 15 Historia de usuario crear ficha técnica

Historia de usuario	
Número: 7	Nombre Historia de Usuario: Crear ficha técnica

ANEXOS

Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad : Alta	Tiempo Estimado: 7 días
Riesgo en Desarrollo: Medio	Tiempo Real: 10 días
Descripción: El sistema permitirá crear las fichas técnicas.	
Observaciones: El sistema mostrará las opción para crear las fichas técnicas correspondientes a la base de cálculo creada.	
Prototipo de interfaz: No aplica	

Tabla 16 Historia de usuario mostrar ficha técnica

Historia de usuario	
Número: 8	Nombre Historia de Usuario: Mostrar ficha técnica
Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad : Alta	Tiempo Estimado: 1 día
Riesgo en Desarrollo: Medio	Tiempo Real: 1 día
Descripción: El sistema permitirá mostrar las fichas técnicas.	
Observaciones: El sistema mostrará las fichas técnicas correspondientes a la base de cálculo creada.	
Prototipo de interfaz: No aplica	

Tabla 17 Historia de usuario modificar ficha técnica

Historia de usuario	
Número: 9	Nombre Historia de Usuario: Modificar ficha técnica
Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad : Alta	Tiempo Estimado: 1 día
Riesgo en Desarrollo: Medio	Tiempo Real: 1 día

Descripción: El sistema permitirá modificar las fichas técnicas.
Observaciones: El sistema mostrará las opción para modificar las fichas técnicas correspondientes a la base de cálculo creada.
Prototipo de interfaz: No aplica

Tabla 18 Historia de usuario eliminar ficha técnica

Historia de usuario	
Número: 10	Nombre Historia de Usuario: Eliminar ficha técnica
Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad : Alta	Tiempo Estimado: 1 día
Riesgo en Desarrollo: Medio	Tiempo Real: 1 día
Descripción: <i>El sistema permitirá eliminar las fichas técnicas.</i>	
Observaciones: El sistema mostrará las opción para eliminar las fichas técnicas correspondientes a la base de cálculo creada.	
Prototipo de interfaz: No aplica	

Tabla 19 Historia de usuario crear base de cálculo

Historia de usuario	
Número: 11	Nombre Historia de Usuario: Crear base de cálculo
Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad : Alta	Tiempo Estimado: 7 días
Riesgo en Desarrollo: Medio	Tiempo Real: 9 días
Descripción: El sistema permitirá crear las base de cálculos que servirán para crear las fichas técnicas correspondientes.	

Observaciones: El sistema mostrará la opción para crear las bases de cálculos que servirán de base para crear las fichas técnicas correspondientes.

Prototipo de interfaz: No aplica

Tabla 20 Historia de usuario mostrar base de cálculo

Historia de usuario	
Número: 12	Nombre Historia de Usuario: Mostrar base de cálculo
Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad : Alta	Tiempo Estimado: 1 día
Riesgo en Desarrollo: Medio	Tiempo Real: 1 día
Descripción: El sistema permitirá mostrar las base de cálculos que servirán para crear las fichas técnicas correspondientes.	
Observaciones: El sistema mostrará la opción para mostrar las bases de cálculos que servirán de base para crear las fichas técnicas correspondientes.	
Prototipo de interfaz: No aplica	

Tabla 21 Historia de usuario modificar base de cálculo

Historia de usuario	
Número: 13	Nombre Historia de Usuario: Modificar base de cálculo
Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad : Alta	Tiempo Estimado: 1 día
Riesgo en Desarrollo: Medio	Tiempo Real: 1 día
Descripción: El sistema permitirá modificar las base de cálculos que servirán para crear las fichas técnicas correspondientes.	
Observaciones: El sistema mostrará la opción para modificar las bases de cálculos que servirán de base para crear las fichas técnicas correspondientes.	

Prototipo de interfaz: No aplica

Tabla 22 : Historia de usuario eliminar base de cálculo

Historia de usuario	
Número: 14	Nombre Historia de Usuario: Eliminar base de cálculo
Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad : Alta	Tiempo Estimado: 1 día
Riesgo en Desarrollo: Medio	Tiempo Real: 1 día
Descripción: El sistema permitirá eliminar las base de cálculos que servirán para crear las fichas técnicas correspondientes.	
Observaciones: El sistema mostrará la opción para eliminar las bases de cálculos que servirán de base para crear las fichas técnicas correspondientes.	
Prototipo de interfaz: No aplica	

Tabla 23 Historia de usuario exportar ficha técnica.

Historia de usuario	
Número: 15	Nombre Historia de Usuario: Exportar Ficha Técnica
Usuario: Susana Calás Jerez	Iteración Asignada: 1
Prioridad : Media	Tiempo Estimado: 7 días
Riesgo en Desarrollo: Medio	Tiempo Real: 8 días
Descripción: El sistema permitirá exportar las fichas técnicas que serán realizadas a partir de las bases de cálculos correspondientes.	
Observaciones: El sistema mostrará la opción para exportar las fichas técnicas correspondientes a partir de un botón que permitirá que la acción se realice fácil y satisfactoriamente.	
Prototipo de interfaz: No aplica	

ANEXOS

1	El sistema de gestión de ofertas comerciales presenta facilidad de configuración	
2	El sistema de gestión de ofertas comerciales permite crear un ficha técnica	
3	El sistema de gestión de ofertas comerciales permite crear una base de cálculo	
4	El sistema de gestión de ofertas comerciales presenta interfaces amigables al usuario	
5	El sistema de gestión de ofertas comerciales permite agilizar el proceso de gestión de la información referente a las ofertas comerciales.	

Valoración

MA -Muy Adecuado (5)

BA – Bastante Adecuado (4)

A - Adecuado (3)

PA – Poco Adecuado (2)

I – Inadecuado (1)