



Universidad de las Ciencias
Informáticas

Facultad 4

**Trabajo de Diploma para optar por el título de Ingeniero
en Ciencias Informáticas**

**Sistema para la gestión de
procesos en la Dirección de
Extensión Universitaria**

Autora: Claudia Rafaela Guilarte Domínguez

Tutores:

Ing. Julio César Espronceda Pérez

Ing. Ortelio Francisco Hernández Socarrás

La Habana, 14 de junio de 2019

FRASE



*Lo importante no es
ser mejor que otros,
sino mejor que ayer*

Jigoro Kano

DECLARACIÓN JURADA DE AUTORÍA

Declaro por este medio que yo Claudia Rafaela Guilarte Domínguez, con carné de identidad 94021840099 soy la autora del trabajo de diploma titulado “*Sistema para la gestión de procesos en la Dirección de Extensión Universitaria*” 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 se firma la presente, a los ____ días del mes de _____ del año 2019.

Autora

Claudia Rafaela Guilarte Domínguez

Tutor

Ing. Julio César Espronceda Pérez

Tutor

Ing. Ortelio Francisco Hernández Socarrás

DATOS DE CONTACTO

Datos de la autora

Nombre y apellidos: Claudia Rafaela Guilarte Domínguez

Correo electrónico: crquilarte@estudiantes.uci.cu

Institución a la que pertenece: Universidad de la Ciencias Informáticas

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

Datos de los tutores

Nombre y apellidos: Julio César Espronceda Pérez

Correo electrónico: jcespronceda@uci.cu

Especialidad de graduación: Ingeniero en Ciencias Informáticas

Año de graduación: 2013

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.

Nombre y apellidos: Ortelio Francisco Hernández Socarrás

Correo electrónico: ofhernandez@uci.cu

Especialidad de graduación: Ingeniero en Ciencias Informáticas

Año de graduación: 2017

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.

DEDICATORIA

Dedico este trabajo a mis padre, por haberme dado la vida y el mejor ejemplo que un hijo puede esperar, por ser la fuente de fuerza e inspiración en mi vida.

A mis abuelos, por su amor, entrega, confianza y darme las fuerzas para seguir adelante.

A mis hermanos, por su cariño y apoyo incondicional.

A mi familia, en general, por estar ahí cuando más los necesitaba y siempre atenta a mi crecimiento como una persona de bien.

A la familia, de mi novío Ortelio, que me han acogido como una integrante más y me han dado su cariño.

Muy especialmente, a Dianellis a quien consideré mi hermana, que aunque hoy no se encuentra entre nosotros para mi fue muy importante, Diane este logro es tuyo también. En fin, a todos lo que han contribuido a que hoy me gradué como Ingeniera en Ciencias Informáticas.

A ustedes, gracias por todo.

Claudia Rafaela Guilarte Domínguez

AGRADECIMIENTOS

Primero que todo quiero agradecer a la Revolución cubana y al Comandante en Jefe Fidel

Castro Ruz creador de esta Universidad en la cual hoy me formo como profesional.

Agradecer también a cada una de las personas que de una forma u otra han aportado su granito de arena en mi formación durante estos años. A todos mis profesores y compañeros que me han ayudado a alcanzar mis metas.

A mis profes Silvia y Rosa, por su constante preocupación y dedicación para que hoy yo lograra mi objetivo.

Al tribunal Coca, Jeiser y Marieta gracias por sus recomendaciones, tiempo y dedicación, así como mi oponente Leduan, a ustedes gracias.

A la primera familia que encontré al llegar a la Facultad Regional de Ciego de Ávila; Alianet, Yanet, Lili, Arlene, Claudia, Doris, Lisandra, Rolando, Yoandi, Alejandro, Camilo, Alain, Karel, Yeider, Yanser, Ramón, Leyriel. Cada uno con su peculiar manera de ser, de una forma u otra me han enseñado algo.

También quiero agradecer a Yailyn, Jessilie, Dalquerine, Daylen, Rene, Pabel, Eric, Liorge, Yenisse, Alexander, Zuyeng y Greter por compartir lindos momentos que nunca olvidaré.

Además, quiero agradecer a los muchachos del grupo por compartir estos años donde nos divertíamos de una forma u otra; a Laurita la mas callada pero incondicional, Arlet, Eliany, Rachel, Karen, Leisdany, Reynaldo, Sander, Eduardo, Jonathan, Anabel, gracias por estar ahí.

No tengo como agradecer a este piquete, familión, como ustedes quieran llamarlos, pero lo que si puedo asegurarles es que una de las cosas mas linda que me ha pasado en la vida.

Lídice la que siempre está peleando pero siempre esta ahí para lo que necesites, ahí y te dice las cosas como son por muy duras que sean, gracias. Tania gracias por estar ahí, ayudarme y brindarme tu cariño. Eliuvis siempre sacándole una sonrisa con sus travesuras o a veces lo contrario cuando se ponía fuerte cuando tena que regañarnos a algunos. Ariel mi contento aunque no le guste que lo llame así, gracias por tus conocimientos y tu amistad incondicional. Carlos gracias por tu apoyo, amistad y confianza. Arlette gracias por soportarnos y arriesgarte a llevarme a estos locos para tu casa a compartir junto a tu familia. Chicos los quiero muchos no tengo manera de cuantificarlo.

A mis tutores por todo su apoyo, tiempo y dedicación: Julio gracias por permitirme trabajar contigo en este Proyecto tan importante para mi y Ortelio gracias por tu entrega incondicional.

A mi novio Ortelio, gracias por tu paciencia, amor, entrega. Gracias por todo gracias por llegar a mi vida.

De igual manera agradecer a la FEU, organización de la cual fui dirigente. Esta me enseñó virtudes como la responsabilidad y el sacrificio. Me permitió conocer a muchos amigos, así como compañeros de trabajo. A, Faire, Yadelis, José Diego, Jesús Miguel, Midiala, Jaier, Luis Miguel, Álvaro, en fin, a toda la familia FEU, gracias.

A mis amigos Esteban, Dayan, Aneylis, Ernesto, Yanet, Reinier, Susana, Leysi, Ewelín y Milena por compartir conmigo durante este tiempo y brindarme su amistad incondicional.

También agradezco a la familia de mi novio que me ha acogido como un miembro más, apoyándome con lo que fuese posible. A todos ustedes gracias.

También quiero aprovechar para agradecer a personas que han sido muy importantes en mi vida Justa, Marcos, el chino, Ismaray, Ramona y Berta quienes me han visto crecer y convertirme en la persona que soy hoy. Gracias por todo.

A Daydee, Elianni, Haydee, Victoria por su amistad incondicional y preocupación.

De manera especial quiero agradecer a mi familia. mis hermanos Ulisito, Mayelin y Joan este título es de ustedes también, a mis tías Carmen, Yole, Mima, Loida, la negra y a mis tíos Raciél, Alexis, Bernando, Amaury, Edin y Mery gracias a todos por siempre velar por mi crecimiento y formación como profesional. A mis primos Laura, Bia, Jose carlos, Aleixto, Milena, Ledaimis, Danay, Ania, Gleyder y mi cuñada Yudeisy gracias por su apoyo. A mis sobrinos por su cariño Melissa, Liscarla y Jean Carlos. El mejor de los agradecimientos va dirigido a los seres que me dieron la vida, por ser las razones de mi existir, a mis padres Teresa y Ulises por su sacrificio, por siempre querer lo mejor para mí y apoyarme en cada decisión, a ustedes mil gracias no existe la forma de pagarles todo lo que ustedes han hecho por mí.

A todas aquellas personas que he conocido, pero no dudan en saludar, preguntarme como me va y se encuentran aquí presentes. Todos y cada uno de ustedes reciban igual hoy mi agradecimiento.

Claudia Rafaela Guilarte Dmínguez

RESUMEN

La Universidad de la Ciencias Informáticas es uno de los centros pertenecientes al Ministerio de Educación Superior (MES). Para su funcionamiento se distribuye en diferentes áreas entre las que se encuentra la Dirección de Extensión Universitaria, la cual realiza diferentes procesos que permiten mejorar el funcionamiento de la misma. Entre los cuales se encuentra la divulgación de actividades, gestión de concursos, reservación de los locales de la Universidad, así como la reservación de actividades externas. A pesar de la gestión que se les realiza a los procesos todavía estos tienen limitantes. Los sistemas relacionados con la Extensión Universitaria, no satisfacen los problemas que existen en la universidad. La presente investigación presenta un sistema para la gestión de procesos de la Dirección de Extensión Universitaria, el cual permite la gestión de los diferentes procesos que se realizan en la Dirección de Extensión Universitaria a partir de varias interfaces que facilitan la automatización de la gestión de dichos procesos. Se presenta una estrategia de pruebas realizada al sistema, con el objetivo de entregar al cliente una solución confiable y libre, que pueda ser empleada como apoyo para un mejor funcionamiento de la Dirección de Extensión Universitaria.

Palabras clave: Dirección de Extensión Universitaria, Extensión Universitaria, procesos, gestión.

ÍNDICE

| | |
|---|-----------|
| INTRODUCCIÓN | 1 |
| CAPÍTULO 1: Marco teórico referencial sobre el Sistema para la gestión de procesos en la Dirección Extensión Universitaria | 7 |
| 1.1. Introducción..... | 7 |
| 1.2. Conceptos asociados a la investigación..... | 7 |
| 1.3. Análisis de las soluciones existentes..... | 8 |
| 1.3.1. Ámbito internacional..... | 8 |
| 1.3.2. Ámbito nacional..... | 9 |
| 1.3.3. UCI..... | 10 |
| 1.4. Entorno de desarrollo de la propuesta de solución..... | 11 |
| 1.4.1. Modelado de software..... | 11 |
| 1.4.2. Herramienta para control de versiones..... | 12 |
| 1.4.3. Lenguaje de Programación (lado del servidor)..... | 12 |
| 1.4.4. Marco de Trabajo..... | 13 |
| 1.4.5. Lenguaje de programación (lado del cliente)..... | 13 |
| 1.4.6. Entorno Integrado de Desarrollo..... | 14 |
| 1.4.7. Sistema Gestor de Base Datos..... | 15 |
| 1.4.8. Servidor Web..... | 15 |
| 1.4.9. Herramientas para pruebas de software..... | 18 |
| 1.4.10. Metodologías de desarrollo de software..... | 21 |
| 1.5. Conclusiones del capítulo..... | 22 |
| CAPÍTULO 2: Características y diseño del Sistema para la gestión de procesos en la Dirección de Extensión Universitaria | 24 |
| 2.1. Introducción..... | 24 |
| 2.2. Análisis..... | 24 |
| 2.2.1 Características del Sistema para la gestión de procesos en la Dirección Universitaria..... | 24 |
| 2.2.2. Mapa Conceptual..... | 25 |

| | |
|---|-----------|
| 2.3. Diseño | 25 |
| 2.3.1. Especificación de requisitos | 26 |
| 2.3.2. Historias de usuario | 29 |
| 2.3.3. Estilo arquitectónico | 30 |
| 2.3.4. Diagrama de clases del diseño | 32 |
| 2.3.6. Modelo de datos | 35 |
| 2.3.7. Modelo de despliegue | 39 |
| 2.4. Conclusiones del capítulo | 40 |
| CAPÍTULO 3: Implementación y pruebas del Sistema para la gestión de procesos en la Dirección de Extensión Universitaria..... | 41 |
| 3.1. Introducción..... | 41 |
| 3.2. Estándar de codificación..... | 41 |
| 3.3. Pruebas de Software | 44 |
| 3.3.1. Pruebas funcionales..... | 44 |
| 3.3.2. Pruebas de carga y estrés | 48 |
| 3.3.3. Pruebas de seguridad | 50 |
| 3.3.4. Pruebas de usabilidad..... | 51 |
| 3.4. Interfaces principales..... | 54 |
| 3.5. Conclusiones del capítulo | 58 |
| CONCLUSIONES GENERALES..... | 59 |
| RECOMENDACIONES..... | 60 |
| REFERENCIAS | 61 |
| ANEXOS..... | 65 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1: Mapa conceptual (Fuente: Elaboración propia) | 25 |
| Figura 2: Funcionamiento del MTV de Django (Infante-Montero, 2012) | 31 |
| Figura 3: Diagrama de clase Crear oferta (Fuente: Elaboración propia)..... | 32 |
| Figura 4: Diagrama de clase Detalles de oferta (Fuente: Elaboración propia)..... | 33 |
| Figura 5 : Modelo de datos (Parte I) (Fuente: Elaboración propia) | 37 |
| Figura 6 Modelo de datos (Parte II) (Fuente: Elaboración propia) | 38 |
| Figura 7: Modelo de despliegue de la solución propuesta (Fuente: Elaboración propia) | 39 |
| Figura 8: Resultados de las pruebas funcionales (Fuente: Elaboración propia) | 47 |
| Figura 9: Pruebas de seguridad, primera iteración (Fuente: Acunetix WVS) | 51 |
| Figura 10: Pruebas de seguridad. Resultado final de la segunda iteración (Fuente: Acunetix WVS).... | 51 |
| Figura 11: Cumplimiento de indicadores de la lista de chequeo de usabilidad (Fuente: Elaboración propia) | 53 |
| Figura 12: Nivel de usabilidad resultante (Fuente: Elaboración propia)..... | 54 |
| Figura 13: Interfaz principal. Portada | 55 |
| Figura 14: Interfaz principal. Reservar oferta. | 56 |
| Figura 15: Interfaz principal. Reservar local. | 57 |
| Figura 16: Interfaz Gestionar publicación | 86 |
| Figura 17: Interfaz Gestionar concursos | 87 |
| Figura 18: Interfaz Gestionar reserva de actividades internas..... | 88 |
| Figura 19: Interfaz Gestionar oferta..... | 89 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla1: Listado de requisitos funcionales | 26 |
| Tabla 2:Historia de usuario Crear oferta | 29 |
| Tabla 3: Historia de usuario Detalles de oferta..... | 30 |
| Tabla 4. Estándares de codificación a utilizar en la implementación del sistema. | 41 |
| Tabla 5. Estrategia de pruebas. | 44 |
| Tabla 6: Variables empleadas en el Caso de Prueba del RF25 Crear oferta..... | 45 |
| Tabla 7:Caso de Prueba del RF25 Crear oferta. | 46 |
| Tabla 8: Resultado de las pruebas de carga y estrés..... | 49 |
| Tabla 9: Resultados de las pruebas de Usabilidad empleando lista de chequeo | 52 |

INTRODUCCIÓN

INTRODUCCIÓN

El Ministerio de Educación Superior (MES) es un sistema integrado por instituciones de educación superior, que rige metodológicamente a las universidades cubanas. Este sistema, bajo la dirección del Partido Comunista de Cuba (PCC) y el Gobierno, forma y consolida valores patrios y de profundo sentido humanista, a la par que preserva, genera y promueve conocimientos, habilidades y competencias, que se reflejan en la formación integral del profesional, la educación posgraduada, las actividades de ciencia, tecnología e innovación y la extensión (MES, 2018).

El MES diseña un Programa de Extensión Universitaria para la Educación Superior Cubana, que tiene la intención de convertirse en un instrumento eficaz para la gestión del proceso extensionista en las universidades. Este programa está concebido con una máxima flexibilidad en su diseño y aplicación, de forma tal, que a partir de la dinámica y las condiciones del entorno. Se pueda buscar un equilibrio coherente, entre las exigencias que plantea el desarrollo de este proceso en la educación superior cubana y las necesidades que demanda cada una de las realidades objeto de transformación por las universidades. El programa tiene como objetivo estratégico: desarrollar la extensión universitaria, transformándola a partir de asumirla como un proceso orientado esencialmente a la labor educativa y político-ideológica, que promueva y eleve la cultura general integral de la comunidad universitaria y de su entorno social (Programa Nacional de Extensión Universitaria, 2018).

Dentro de los centros de estudios pertenecientes al MES se encuentra la Universidad de las Ciencias Informáticas que tiene como misión la formación de profesionales comprometidos con su patria y altamente calificados a través de un modelo de formación caracterizado por el vínculo estudio-trabajo, la realización de tareas investigativas y de carácter extensionistas.

Para su funcionamiento se distribuye en diferentes áreas entre las que se encuentra la Dirección de Extensión Universitaria, la cual propicia y realiza acciones entre la comunidad

INTRODUCCIÓN

universitaria y las localidades cercanas a la institución, posibilitando un trabajo en conjunto. Logrando así preservar y desarrollar la cultura, como necesidad de la sociedad.

Para lograr dicho objetivo, esta área realiza diferentes procesos que permiten mejorar el funcionamiento de la misma. Entre los cuales se encuentra la divulgación de actividades, gestión de concursos, reservación de los teatros de la Universidad (Docente José Martí y Camilo Cienfuegos), así como la reservación de actividades externas. A pesar de la gestión que se les realiza a los procesos todavía estos tienen limitantes en la centralización y organización de la información que se gestiona.

La divulgación que se realiza en la DEU tiene como objetivo dar a conocer a la comunidad universitaria las actividades que se estarán realizando, así como el lanzamiento de concursos logrando así que la información esté al alcance de todos. La forma en que se dan a conocer no es la más efectiva debido a que los medios de comunicación no siempre promocionan las actividades a tiempo o no se le da la publicidad necesaria para que llegue de la forma más amena, lo cual trae como consecuencia que en algunas ocasiones las actividades no tengan la asistencia requerida, lo mismo ocurre con la gestión de concursos o las peñas (música, danza, literatura, cine) provocando que la participación sea escasa.

Por otra parte, la reservación de teatros de la Universidad se realiza a través de una solicitud que efectúa un área en específico con el objetivo de desarrollar diferentes actividades. Para acometer dicha actividad el encargado debe llenar una planilla y mandarla por el correo para pedir la reservación de un teatro determinado donde se debe tener en cuenta la fecha y hora de la reservación para que no coincida con otra solicitud o en el caso de que una actividad tiene más prioridad según su magnitud o que dicha solicitud no sea vista a tiempo trae consigo que las actividades programadas no salgan con la calidad requeridas y en la fecha programada.

INTRODUCCIÓN

De igual manera la Dirección de Extensión Universitaria gestiona reservaciones de actividades externas como son Teatros de la ciudad, Galas, Conciertos, Shows humorísticos; propiciando a la comunidad universitaria variadas ofertas recreativas para un sano esparcimiento. Una vez ofertadas las actividades los interesados deben dirigirse al Docente Mariana Grajales para solicitar la reservación teniendo que salir de su centro laboral para luego recoger las entradas lo que hace engorrosa para el usuario debido al tiempo que debe emplear para dicha gestión además de que puede suceder que se hayan agotado las entradas de esta oferta.

En esta área se evidencian limitantes en la centralización y organización de la información que se gestiona, donde predominan las mencionadas a continuación: toda su información no se encuentra almacenada en un servidor documental centralizado, lo cual puede ocasionar la pérdida de los documentos y provocaría no tener evidencias de las diferentes actividades que se realizan en esta área. Además, no posee una política de acceso a la información pues los documentos no están almacenados de manera segura y estructurada al no contar con la definición de diferentes permisos de acceso a los documentos. No existe un control de cambios, de revisiones y de accesos que se les realicen a los documentos para salvaguardar el valor documental. El flujo de información que se realiza es de forma manual ocasionando que existan retrasos en la entrega de los documentos y no se conoce en qué paso del proceso se encuentra detenida la información.

Por otra parte, existe dependencia del correo para la circulación de la documentación entre las diferentes áreas, provocando que se afecte la entrega de la información. Los documentos, tanto en papel como en soportes digitales, no se encuentran clasificados ni organizados por lo que una búsqueda sobre ellos consume mucho tiempo y esfuerzo por el personal de cada una de estas áreas, además de que dichos documentos no cumplen actualmente los principios de la gestión documental (El blog del hogar, 2016).

INTRODUCCIÓN

La gestión documental eficaz tiene en cuenta los siguientes principios para su funcionamiento son el almacenamiento, la recuperación, la clasificación, la seguridad, la custodia, la distribución, las reglas del workflow, la creación de documentos y la autenticación de los documentos.

Ante estas dificultades, se hace fundamental la necesidad de organizar los documentos de manera funcional, precisa y uniforme durante todo su ciclo de vida: la generación, la revisión y aprobación del mismo, la distribución y, por último, la custodia y disposición; pues a mayor crecimiento de información, mayor es el caos que esta genera independientemente del tipo de organización. Esto se convierte en un factor determinante para la optimización de los procesos en la organización y enfatiza en la necesidad de encontrar una vía que permita organizar y controlar el trabajo que comprende el proceso de gestión de los documentos.

A partir de la situación problemática descrita anteriormente se plantea como **problema de investigación** ¿Cómo contribuir a la organización y centralización de la información en la gestión de procesos de la Dirección de Extensión Universitaria?

El **objetivo** de este trabajo de diploma es desarrollar un sistema para la gestión de procesos en la Dirección de Extensión Universitaria que contribuya a la organización y centralización de la información.

Para cumplir el objetivo se plantean las siguientes **tareas de la investigación**:

1. Análisis de los principales conceptos y trabajos relacionados con los sistemas de gestión de la información.
2. Análisis del entorno para el desarrollo del sistema.
3. Estudio de la metodología y las tecnologías a emplear en el proceso de desarrollo de software.
4. Análisis y diseño del sistema para la gestión de procesos.

INTRODUCCIÓN

5. Implementación del sistema para la gestión de procesos en la dirección de extensión universitaria.
6. Prueba del sistema para la gestión de procesos en la dirección de extensión universitaria.

La investigación en curso se enmarca en el **objeto de estudio**: gestión de procesos extensionistas, delimitando como **campo de acción**: la gestión de los procesos en la Dirección de Extensión Universitaria en la UCI.

Los **métodos teóricos** permiten estudiar las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e hipótesis de investigación; de ellos se emplearon los siguientes:

Histórico-Lógico: se emplea para identificar posibles funcionalidades que pueda tener el sistema a partir del análisis de la evolución de los sistemas de gestión para procesos de extensión universitaria.

Análisis-Síntesis: empleado para el análisis, evaluación y selección de las técnicas a emplear en el desarrollo del sistema. Así como para sintetizar la información que se obtuvo mediante la entrevista con el cliente de manera que pudiera ser usada en el desarrollo del mismo, además, en la identificación de los elementos del marco teórico de la investigación.

Modelación: para realizar una representación del proceso estudiado que sirva de guía en el desarrollo del sistema, y mediante este, identificar las características y relaciones fundamentales.

Los **métodos empíricos** representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a determinada elaboración racional; de ellos se empleo el siguiente:

Entrevista: empleado en los encuentros con el cliente para obtener la información necesaria

INTRODUCCIÓN

que permita determinar las características, cualidades y requisitos con los que debe contar la propuesta de solución.

Observación: empleado para obtener el conocimiento necesario del funcionamiento de las soluciones existentes para la gestión de procesos de la Dirección de Extensión Universitaria.

Análisis Documental: es un conjunto de operaciones encaminadas a representar un documento y su contenido bajo una forma diferente de su forma original, con la finalidad de posibilitar su recuperación posterior e identificarlo.

El presente trabajo de diploma está estructurado de la siguiente manera: introducción, tres capítulos, conclusiones generales, recomendaciones y referencias bibliográficas empleadas durante el desarrollo de la investigación, y por último para complementar la investigación se presentan una serie de anexos. A continuación, se muestra la descripción de los capítulos.

Capítulo 1. Marco teórico referencial sobre el Sistema para la gestión de procesos en la Dirección Extensión Universitaria: se realiza un estudio del arte sobre los principales presupuestos de la investigación. Se describen el proceso de desarrollo de software, así como las tendencias, técnicas, metodología y tecnologías usadas en la propuesta de solución.

Capítulo 2. Características y diseño del Sistema para la gestión de procesos en la Dirección de Extensión Universitaria: se identifican y describen los conceptos asociados al dominio del problema y los procesos relacionados con el negocio teniendo en cuenta la metodología seleccionada en el capítulo anterior. Se definen cuáles son los requerimientos funcionales, no funcionales y el modelo de datos.

Capítulo 3. Implementación y pruebas del Sistema para la gestión de procesos en la Dirección de Extensión Universitaria: en este capítulo se define el estándar de codificación que sirve de guía para la implementación de la solución propuesta, así como la estrategia de pruebas a aplicar para lograr un correcto funcionamiento.

CAPÍTULO 1: Marco teórico referencial sobre el Sistema para la gestión de procesos en la Dirección Extensión Universitaria

1.1. Introducción

Durante una investigación es necesario obtener conocimiento teórico que facilite la realización de las actividades y así poder llevar a la práctica lo aprendido. Para ello se hace necesario realizar una profunda búsqueda bibliográfica con el fin de lograr una mayor comprensión del alcance de la investigación. En este capítulo se abordan los elementos teóricos que fundamentan la presente investigación. Se realiza un análisis de los sistemas existentes relacionados para la gestión de procesos de extensión universitaria. Se incluye además un análisis de la metodología, herramientas y tecnologías a emplear en el desarrollo de la propuesta de solución.

1.2. Conceptos asociados a la investigación

Para ayudar a entender el desarrollo de la investigación, se relacionarán a continuación los conceptos principales que sirven de soporte.

La **Extensión Universitaria** constituye uno de los procesos medulares de la educación superior en Cuba y está enfocada a la reafirmación de la identidad cultural, la formación de valores y hacia una mejor calidad de vida de los miembros de la comunidad universitaria.

La cual cuenta con la **Dirección de Extensión Universitaria** es el órgano rector del proceso integrador y dinamizador del flujo cultural y la comunicación entre la universidad y la sociedad, así como también entre cada uno de sus miembros.

La DEU en la UCI dentro de sus líneas de trabajo cuenta con la creación de proyectos socioculturales comunitarios con el fin de estrechar cada vez más los lazos de integración de la universidad con la sociedad, además de impulsar la creación y la promoción del arte como

un medio en el que estudiante se sienta y comporte como universitario en el plano de su disfrute cultural e ideológico, con un alto nivel de sensibilidad y de identidad con su centro.

Para analizar dichos problemas nos enfocaremos en los procesos que se manejan en la DEU a través de la **gestión** la cual proviene del latín *gestio*, el concepto de gestión hace referencia a la acción y a la consecuencia de administrar o gestionar algo. Al respecto, hay que decir que gestionar es llevar a cabo diligencias que hacen posible la realización de una operación comercial o de un anhelo cualquiera. Administrar, por otra parte, abarca las ideas de gobernar, disponer, dirigir, ordenar u organizar una determinada cosa o situación. La noción de gestión, por lo tanto, se extiende hacia el conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto. La gestión es también la dirección o administración de una compañía o de un negocio (Pérez Porto & Merino, 2012).

Un **proceso** es una secuencia de pasos dispuesta con algún tipo de lógica que se enfoca en lograr algún resultado específico (Definición, 2018).

1.3. Análisis de las soluciones existentes

Actualmente existen diversas soluciones para la gestión de procesos. Pero en el ámbito de la presente investigación, se tomaron en cuenta solamente aquellas que permitieran la gestión de procesos de extensión. A continuación, se relacionan sus características y principales funcionalidades.

1.3.1. Ámbito internacional

Sistema para la Prevención de la Deserción en la Educación Superior

El Ministerio de Educación Nacional de Colombia propuso disminuir la deserción estudiantil en la educación superior como parte de la estrategia planteada para aumentar la cobertura, la calidad y la eficiencia educativas. Entre otras acciones, puso en marcha el Sistema para la Prevención de la Deserción en la Educación Superior —Spadies— ante la necesidad de contar con una visión sectorial e integrada de la problemática de la deserción, a partir de la

cual se pudiera disponer de una conceptualización, una medición y una metodología de seguimiento del fenómeno aplicables a todas las instituciones de educación superior de ese país (CAN, 2012).

La información que el sistema provee se encuentra disponible al público mediante un módulo de consulta on-line y un software de aplicación local en cada Instituto de Educación Superior. Este sistema le permite al Ministerio medir y monitorear los factores determinantes de la deserción, conocer su evolución en el tiempo y ver cómo se comportan diferentes instituciones y regiones. Igualmente, hace posible que cada institución cuente hoy con un perfil de sus estudiantes y con sistemas de alertas tempranas sobre los factores que los hacen vulnerables, lo que sirve para orientar de forma más eficiente apoyos y políticas. Además, permite hacer seguimiento estadístico a los niveles de deserción, seguimiento a las condiciones de ingreso de los estudiantes, seguimiento al comportamiento de factores determinantes del fenómeno, estimación del riesgo de deserción para cada estudiante. Facilita la elección y evaluación de estrategias institucionales de apoyo a los estudiantes.

Esta puede realizarse con relación al agregado nacional, a grupos de instituciones, a grupos de programas académicos. Brinda información confiable y actualizada a medios de comunicación y a la comunidad educativa en general sobre el estado de la deserción estudiantil en Colombia (CAN, 2012).

1.3.2. Ámbito nacional

Página web extensión universitaria de la Universidad de Holguín

Esta página web facilita la integración, la divulgación de actividades, es de fácil acceso para el público la cual tiene como objetivo mantener informados al estudiantado y al claustro docente. También se muestran los resultados alcanzados hasta la actualidad (Quiñones Laffita, Hernández Torres, & Córdón González, 2018).

Sistema informático de gestión universitaria (SIGUA)

Esta aplicación web fue diseñada con el objetivo final de facilitar el trabajo en los diferentes departamentos y secretarías docentes, al igual que a los estudiantes en el Instituto Superior de Arte. La interfaz Web que brinda esta aplicación permite agregar nuevos datos y realizar consultas a los mismos. La información introducida se almacena en la base de datos modelada para este fin (Guillot Jiménez, 2014).

1.3.3. UCI

Portal Intranet

Este portal brinda información a la comunidad universitaria de las diferentes áreas de la universidad, entre ellas se encuentra extensión, donde hace referencia a los Proyectos extensionistas, Movimiento artístico y recreación, Bienestar Universitario, Patrimonio cultural, Librería, Residencia y Cátedras honoríficas. Además, muestra la cartelera cultural de la semana que ofrece la Universidad, así como actividades que se desarrollarán fuera de esta.

Sistema de gestión universitaria

Este sistema gestiona información acerca de las áreas docentes, productivas e investigativas de la universidad con respecto a la Residencia, Cooperación, Biblioteca, Investigación, Tecnologías, Teleformación, Ingreso y Egreso, así como Extensión, Pregrado y Postgrado. El cual fue desarrollado por la Dirección de Informatización perteneciente a la Vicerrectoría de nuestra universidad.

Conclusiones del análisis de las soluciones existentes

Los sistemas estudiados relacionados con la Extensión Universitaria, no satisfacen los problemas que existen en la universidad, como son la divulgación de la información, que no existe control y organización sobre la misma. No gestionan la reservación de los teatros de la universidad y ni las actividades externas, en su mayoría realizadas en los teatros de La Habana. Por lo tanto, constituye una necesidad primordial desarrollar el sistema para la gestión de procesos de la Dirección de Extensión Universitaria. Además, el estudio de estos

sistemas permitió estudiar las nuevas tendencias de la arquitectura de la información. También posibilitó analizar las tecnologías empleadas en el desarrollo de cada sistema.

1.4. Entorno de desarrollo de la propuesta de solución

1.4.1. Modelado de software

El modelado de software es una técnica para tratar con la complejidad inherente a estos sistemas. El uso de modelos ayuda al ingeniero de software a "visualizar" el sistema a construir. Además, los modelos de un nivel de abstracción mayor pueden utilizarse para la comunicación con el cliente. Por último, las herramientas de modelado y las de Ingeniería de Software Automatizada pueden ayudar a verificar la corrección del modelo (Academic, 2012).

Lenguaje Unificado de Modelación (UML)

Como lenguaje de modelado se utiliza el Lenguaje Unificado de Modelado (*UML* por sus siglas en inglés) en su versión 2.1. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Incluye aspectos conceptuales tales como procesos de negocio y funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables (García, 2005).

Herramienta para el modelado

Para la presente investigación se utiliza la herramienta de modelado Visual Paradigm 8.0, es una herramienta de Ingeniería Asistida por Computadora (*CASE*¹ por sus siglas en inglés), que está diseñada para modelar un sistema de información empresarial y gestionar procesos de desarrollo. Esta herramienta soporta los principales lenguajes y estándares de modelado de la industria, en los que incluye el Lenguaje de Modelado Unificado (UML). Ofrece herramientas completas para la captura de requisitos de software, análisis de procesos,

¹*CASE* (*Computer Aided Software Engineering*) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software (Marqués-Andrés, 2001).

modelado de software, entre otras (Visual Paradigm, 2016).

1.4.2. Herramienta para control de versiones

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que se pueda recuperar versiones específicas más adelante (Saldaña, 2017).

Para garantizar el control de versiones en el sistema, se propone el uso de la versión 2.8 del cliente Git para la plataforma Gitlab disponible en la UCI en la dirección <http://codecomunidades.prod.uci.cu>, el cual, según Saldaña (2017):

- Es un sistema de control de versiones distribuido
- No depende de acceso a la red o un repositorio central
- Está enfocado a la velocidad, uso práctico y manejo de proyectos grandes

1.4.3. Lenguaje de Programación (lado del servidor)

Según (Hernandez Castillo, 2008), un lenguaje de programación consiste en un conjunto de órdenes o comandos que describen un proceso determinado. Cada lenguaje tiene sus instrucciones y enunciados verbales propios, que se combinan para formar los programas de cómputo. Los lenguajes de programación no son aplicaciones, sino herramientas que se pueden usar para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

Como lenguaje del lado del servidor se empleó Python en su versión 3.6.1, el cual es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, debido que soporta orientación a objetos, programación imperativa y en menor medida, programación

funcional. Es uno de los lenguajes de programación más populares en los últimos años debido a la curva de aprendizaje (Rossum, 2017).

1.4.4. Marco de Trabajo

Un marco de trabajo o framework es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Proporcionan un conjunto de librerías con funcionalidades que aumentan la facilidad del trabajo y disminuyen su complejidad. Además, permiten reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones (Euphoria IT, 2008). En la actualidad son muy usados para simplificar y agilizar el proceso de desarrollo de aplicaciones.

Para el desarrollo del sistema se utiliza el framework Django en su versión 2.1.7, que es un marco de trabajo de desarrollo web basado en la arquitectura Model-View-Template que solo cambia los nombres del estilo clásico Modelo-Vista-Controlador (MVC) y les otorga a las vistas toda la responsabilidad del negocio. El mismo está totalmente implementado sobre Python, con el que se pueden crear y mantener aplicaciones de alta calidad e incluye un servidor web ligero que se puede usar mientras se desarrolla (Django, 2017).

1.4.5. Lenguaje de programación (lado del cliente)

En el desarrollo del sistema se utiliza el lenguaje de programación para el lado del cliente JavaScript. Este, es un lenguaje interpretado que permite incluir macros en páginas web. Estas macros se ejecutan en el ordenador del visitante, y no en el servidor, pues los servidores web suelen estar sobrecargados, mientras que las computadoras de los usuarios no suelen estarlo (Gutiérrez, 2009).

Biblioteca

Una biblioteca es un conjunto de recursos (algoritmos) prefabricados, que pueden ser utilizados por el programador para realizar determinadas operaciones. Las declaraciones de las funciones utilizadas en estas bibliotecas, junto con algunas macros y constantes

predefinidas que facilitan su utilización, se agrupan en ficheros de nombres conocidos que suelen tener las extensiones lib, bpl, a, dll, js, entre otras (Torrealba, 2016). La biblioteca utilizada en el desarrollo del sistema es:

JQuery, es una biblioteca multiplataforma de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML², manipular el árbol DOM³, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX⁴ a páginas web. Es además un software libre y de código abierto, permitiendo su uso en proyectos tanto libres como privados. JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio (Gutiérrez, 2009).

1.4.6. Entorno Integrado de Desarrollo

Para agilizar la construcción de aplicaciones, los desarrolladores se apoyan de un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés). Estos programas contienen un conjunto de herramientas de programación que permiten el desarrollo de otras aplicaciones en determinado lenguaje. Estos pueden realizar las funciones de un editor de código, un compilador, depurador y hasta un constructor de interfaz gráfica (Vigo, 2014).

En el desarrollo del sistema se emplea el IDE JetBrains PyCharm en su versión 2018.3.4, el cual es un entorno de desarrollo integrado de código abierto y multiplataforma desarrollado por JetBrains que se utiliza para programar en Python. Proporciona un análisis de código, depuración gráfica, probador de unidad integrada y apoya el desarrollo web con el framework Django (JetBrains Pycharm, 2017).

² Del inglés *HyperText Markup Language* (Lenguaje de marcas de hipertexto).

³ Del inglés *Document Object Model* (Modelo de Objetos del Documento o Modelo en Objetos para la Representación de Documentos).

⁴ Del inglés, acrónimo de JavaScript asíncrono y XML.

1.4.7. Sistema Gestor de Base Datos

Un Sistema Gestor de Base de Datos (SGBD, en inglés DBMS: Data Base Management System) es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarias para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejarlos (Ramos, 2014).

Para garantizar la persistencia de los datos generados por el sistema, es empleado el SGBD PostgreSQL versión 10, el cual está orientada a objetos y libre, publicado bajo la licencia BSD. Este potente gestor implementa el estándar SQL92/SQL99 y soporta distintos tipos de datos, tipos fecha, monetarios, elementos gráficos y cadenas de bits. Permite la creación de tipos propios e incorpora una estructura de datos array (PostgreSQL Global Development Group, 2014).

1.4.8. Servidor Web

Servidor Web es un programa que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un Navegador Web. Para la transmisión de todos estos datos se utiliza algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación de la arquitectura TCP (Apache Software Foundation, 2011).

Servidor WSGI HTTP para Python

Como servidor HTTP para Python se empleó **Gunicorn** o también conocido como “*Green*”

Unicorn”, el cual es un servidor HTTP WSGI de Python para UNIX. Es ampliamente compatible con varios *framework* web, de implementación simple, rápido y ligero en cuanto a los recursos del servidor. Permite administrar las peticiones simultaneas que la aplicación reciba y el número de peticiones simultáneas que se pueden utilizar se indica mediante el atributo *workers* (Chesneau, 2018).

Sistema de control de procesos UNIX

Para administrar la ejecución del servidor HTTP se empleó **Supervisor**, que no es más que un sistema cliente / servidor que permite a sus usuarios monitorear y controlar una serie de procesos en los sistemas operativos UNIX. Supervisor está destinado a ser utilizado para el control de los procesos relacionados con un proyecto o un cliente, y está destinado a iniciar como cualquier otro programa en el arranque (McDonough, 2018).

Los procesos a menudo necesitan ser iniciados y detenidos en grupos, a veces incluso en una “orden de prioridad”. Supervisor permite asignar prioridades a los procesos e iniciarlos teniendo en cuenta estas. Además, los procesos se pueden agrupar en “grupos de procesos” y un “conjunto de procesos relacionados lógicamente” pueden ser detenidos e iniciados como una unidad o un solo proceso (McDonough, 2018).

Servidor web/proxy inverso Nginx

Para servir archivos estáticos y funcionar como proxy inverso a las peticiones al servidor WSGI HTTP se utilizó **Nginx**. Este es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico. Es software libre y de código abierto, licenciado bajo la Licencia BSD simplificada; también existe una versión comercial distribuida bajo el nombre de Nginx Plus. Es multiplataforma, por lo que corre en sistemas tipo Unix (GNU/Linux, BSD, Solaris, Mac OS X, entre otros) y Windows (NGINX Inc, 2017).

A diferencia de los servidores tradicionales, Nginx no se basa en subprocesos para manejar las solicitudes. En su lugar, utiliza una arquitectura mucho más escalable basada en eventos

(asíncrona). Esta arquitectura utiliza cantidades pequeñas, pero más importantes, predecibles de memoria bajo carga. Incluso si no espera manejar miles de solicitudes simultáneas, aún puede beneficiarse del alto rendimiento de Nginx y la pequeña huella de memoria. Nginx se puede escalar en todas las direcciones: desde el VPS más pequeño hasta grandes grupos de servidores (NGINX Inc, 2017).

Otras características que ofrece el servidor Nginx son:

1. Capaz de manejar más de 10.000 conexiones simultáneas con un uso bajo de memoria.
2. Balanceo de carga, distribuye la carga entre los servidores que formen parte de la estructura, redirigiendo cada vez la petición hacia aquella máquina que tenga una menor carga.
3. Alta tolerancia a fallos.
4. Soporte para TSL, SSL, FastCGI, SCGI o uWSGI, entre otros.
5. Compatible con el nuevo estándar de direcciones IPv6

Nginx permite ampliar su funcionalidad por medio del uso de módulos. Algunos de los módulos más importantes que se puede encontrar son (Acens, 2013):

1. HTTP Referer: permite filtrar peticiones recibidas en función de la cabecera Referer.
2. HTTP Limit Zone: limita el número de conexiones simultáneas desde un mismo cliente.
3. User ID: proporciona cookies identificativas.
4. FLV: permite reproducir vídeo en streaming.
5. Perl: módulo que permite ejecutar Perl directamente dentro de Nginx y llama a Perl a través de SSI.
6. WebDAV: ofrece soporte para WebDAV
6. Secure Link: este módulo ofrece la posibilidad de proteger páginas mediante clave secreta.

7. XSLT: funcionalidad que permite el post-procesamiento de páginas mediante XSLT.

Se utilizará este servidor web porque el sistema para la gestión de procesos de la Dirección de Extensión Universitaria se desarrollará entorno a la universidad, la cual usa este servidor para alojar los sitios web internos de la comunidad universitaria.

1.4.9. Herramientas para pruebas de software

Las pruebas de software, son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada (Pressman, 2010).

En el caso de la propuesta de solución de la investigación, este será validado a través de varios tipos y técnicas de pruebas, algunas de ellas de forma manual y otras mediante el uso de herramientas que permiten la realización de dicha tarea. Entre las herramientas existentes, se pretende utilizar las herramientas Acunetix Web Vulnerability Scanner y Apache JMeter, para las pruebas de seguridad y de carga y estrés, respectivamente.

Los firewalls, SSL y las redes reforzadas son inútiles contra el pirateo de aplicaciones web. Los piratas informáticos se concentran en aplicaciones basadas en la web (carritos de compras, formularios, páginas de inicio de sesión, etc.), accesibles las 24 horas del día, los 7 días de la semana, y están conectados directamente a los fondos de su base de datos con información valiosa. Las aplicaciones web están hechas a la medida, menos probadas que el software comercial y es probable que tengan vulnerabilidades no descubiertas que pueden ser una receta para el desastre (Acunetix, 2018).

Acunetix es el escáner de vulnerabilidad web líder utilizado por compañías serias de Fortune 500 y ampliamente reconocido por incluir la inyección de SQL más avanzada y la tecnología de escaneo de caja negra XSS. Rastrea automáticamente sus sitios web y realiza técnicas de hacking de caja negra Y caja gris que encuentran vulnerabilidades peligrosas que pueden comprometer su sitio web y sus datos (Acunetix, 2018).

CAPÍTULO 1

Acunetix realiza pruebas de Inyección SQL, XSS, XXE, SSRF, Inyección de encabezado de host y más de 4500 vulnerabilidades web. Cuenta con las más avanzadas técnicas de escaneo generando los menos falsos positivos posibles. Simplifica el proceso de seguridad de la aplicación web a través de sus características de administración de vulnerabilidades incorporadas que lo ayudan a priorizar y administrar la resolución de vulnerabilidades (Acunetix, 2018).

- Análisis y rastreo en profundidad: analiza automáticamente todos los sitios web
- La mayor tasa de detección de vulnerabilidades con bajos falsos positivos
- Gestión integrada de vulnerabilidades: prioriza y controla las amenazas
- Integración con WAFs populares y rastreadores de problemas
- Herramientas gratuitas de escaneo de seguridad de red y pruebas manuales
- Disponible en Windows, Linux y en línea.

Acunetix es un punto clave en la estrategia de seguridad de nuestra aplicación, se integra con el proceso de control de calidad, lo que nos permite una forma rentable de detectar fallas que se pueden resolver de manera temprana dentro del ciclo de vida del desarrollo. Su caja de herramientas también facilita la realización de pruebas manuales. Después de probar muchos otros, podemos decir que es el más rápido y tiene la mejor relación entre los hallazgos y los falsos positivos (Acunetix, 2018).

La aplicación Apache JMeter [™] es un software de código abierto, una aplicación Java 100% pura diseñada para cargar el comportamiento funcional de la prueba y medir el rendimiento. Originalmente fue diseñado para probar aplicaciones web, pero desde entonces se ha expandido a otras funciones de prueba (Apache JMeter, 2018).

Apache JMeter se puede usar para probar el rendimiento tanto en recursos estáticos como dinámicos, aplicaciones dinámicas web. Se puede usar para simular una carga pesada en un servidor, grupo de servidores, red u objeto para probar su resistencia o para analizar el

rendimiento general bajo diferentes tipos de carga (Apache JMeter, 2018).

Las características de Apache JMeter incluyen:

- Capacidad de carga y prueba de rendimiento de muchos tipos diferentes de aplicaciones / servidor / protocolo:
 - Web: HTTP, HTTPS (Java, NodeJS, PHP, ASP.NET, ...)
 - Servicios Web de SOAP / REST
 - FTP
 - Base de datos a través de JDBC
 - LDAP
 - Middleware orientado a mensajes (MOM) a través de JMS
 - Correo: SMTP (S), POP3 (S) e IMAP (S)
 - Comandos nativos o scripts de shell
 - TCP
 - Objetos Java
- IDE de prueba con todas las funciones que permite la grabación rápida del plan de prueba (desde navegadores o aplicaciones nativas), compilación y depuración.
- Modo CLI (modo de línea de comandos (anteriormente denominado no GUI) / modo sin cabeza) para cargar la prueba desde cualquier SO compatible con Java (Linux, Windows, Mac OSX, ...)
- Un informe HTML completo y listo para presentar.
- Correlación sencilla mediante la capacidad de extraer datos de los formatos de respuesta más populares, HTML, JSON, XML o cualquier formato de texto
- Portabilidad completa y pureza 100% Java.
- El marco completo de subprocesos múltiples permite el muestreo simultáneo mediante varios subprocesos y el muestreo simultáneo de diferentes funciones mediante grupos de subprocesos separados.
- Caché y análisis / reproducción fuera de línea de resultados de pruebas.

- Núcleo altamente extensible:
 - Los muestreadores enchufables permiten capacidades de prueba ilimitadas.
 - Muestreadores de secuencias de comandos (lenguajes compatibles con JSR223 como Groovy y BeanShell)
 - Se pueden elegir varias estadísticas de carga con temporizadores conectables.
 - Los complementos de análisis y visualización de datos permiten una gran extensibilidad y personalización.
 - Las funciones se pueden utilizar para proporcionar una entrada dinámica a una prueba o para la manipulación de datos.
 - Fácil integración continua a través de bibliotecas de código abierto de terceros para Maven, Gradle y Jenkins.

1.4.10. Metodologías de desarrollo de software

Como define Méndez (2010), una metodología de desarrollo de software es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo producto de software. Puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto.

AUP-UCI

Para el desarrollo de la solución se emplea la variación de la metodología de Proceso Unificado Ágil (AUP por sus siglas en inglés), por ser la empleada como estándar para guiar el proceso productivo de la UCI.

La metodología AUP contribuye de manera significativa al desarrollo del software que se espera obtener, ya que esta metodología además de ser ágil es también adaptable a cualquier circunstancia que pueda surgir en el proceso de desarrollo del proyecto (Ambler, 2014).

La UCI le ha realizado modificaciones con el fin de adaptarlo al ciclo de vida definido para la actividad productiva de dicha institución; de las 4 fases que encierra la metodología AUP se simplificaron a (Sánchez, 2015):

- Inicio: esta etapa se mantiene de AUP, pero se modifica el objetivo de la misma; realización de la planeación del proyecto en cuanto a un estudio de la organización del cliente, el alcance del proyecto, estimaciones de tiempo esfuerzo y costo que serán necesarios para decidir si se desarrolla el producto o no.
- Ejecución: en esta fase se recogen las actividades que desarrolla AUP de elaboración, construcción y transición. Se ejecutan las actividades requeridas para obtener el software y ajustar los planes del proyecto, realizando las tareas de modelado del negocio, obtención y análisis de los requisitos, elaboración de la arquitectura, diseño, implementación y liberación de producto.
- Cierre: en esta fase se analizan los resultados del proyecto, su ejecución y se llevan a cabo actividades formales de cierre del proyecto.

1.5. Conclusiones del capítulo

Durante este capítulo se trataron los elementos teóricos que sustentan a la propuesta de solución del problema planteado, en este sentido se concluye:

- La sistematización de los conceptos fundamentales asociados al dominio de la presente investigación y sus relaciones, proporcionó una mayor comprensión de la propuesta de solución.
- El análisis de los sistemas homólogos existentes, permitió arribar a la conclusión que no se ajustan completamente a las necesidades de la propuesta de solución, pero si aportó una visión más amplia del objeto de estudio de la investigación. Además, posibilitó adquirir y fomentar los conocimientos necesarios para el desarrollo del sistema.

CAPÍTULO 1

- El estudio de las tecnologías de la informática y las comunicaciones y la observación de las distintas tendencias en el desarrollo de software, permitió identificar al Django como framework de desarrollo, PostgreSQL como sistema gestor de bases de datos, Python como lenguaje de programación del lado del servidor, y como lenguaje del lado del cliente JavaScript mediante la librería JQuery.

CAPÍTULO 2

CAPÍTULO 2: Características y diseño del Sistema para la gestión de procesos en la Dirección de Extensión Universitaria

2.1. Introducción

Establecer una estrecha relación entre el cliente y el equipo de trabajo en función lograr los objetivos, es una de las prioridades cuando se desea desarrollar software. El presente capítulo aborda los principales aspectos relacionados con las características de la propuesta de solución. Donde se define el modelo conceptual, según el objeto de estudio, haciendo uso de la metodología AUP en su variante para guiar el proceso productivo de la UCI. Se identifican los requisitos funcionales y no funcionales con los que debe cumplir la propuesta de solución, así como estilo arquitectónico y los patrones de diseño para lograr buenas prácticas en el diseño y posterior implementación de la solución propuesta. Igualmente se muestran los principales artefactos de ingeniería de software propuestos por la metodología AUP-UCI correspondientes a las funcionalidades.

2.2. Análisis

En esta fase se presentan las principales características asociadas a la propuesta de solución. Se realiza, además, un análisis de los principales conceptos asociados al dominio de la presente investigación evidenciando su relación por medio de un mapa conceptual.

2.2.1 Características del Sistema para la gestión de procesos en la Dirección Universitaria

Dadas las necesidades planteadas en la situación problemática de la presente investigación, la solución propuesta constituye un sistema para la gestión de procesos de la Dirección de Extensión Universitaria.

Este sistema permite divulgar las diferentes ofertas recreativas propuestas por la universidad a desarrollarse dentro o fuera de la misma; para ello el usuario realiza su reservación a través del sistema y después cuando este se autentique, deberá entrar a la sesión de Ofertas y ahí podrá ver si su solicitud fue aceptada. En cuanto a la gestión de las reservas de los diferentes locales del centro una vez realizada la solicitud de reservación del local, el gestor de locales confirmará o no su solicitud atendiendo la disponibilidad del local teniendo en cuenta la

CAPÍTULO 2

importancia del evento a realizar. Los usuarios pueden participar en los concursos que se muestren el sistema, además de poder comentar en la publicación de su interés. Además, el sistema brindará información acerca de los cursos electivos que oferta la Dirección de Extensión Universitaria, así como los Proyectos Extensionistas y Cátedras Honoríficas que existen en la Universidad.

A lo largo del desarrollo del capítulo se describen los artefactos que permiten conceptualizar y comprender mejor el funcionamiento del sistema.

2.2.2. Mapa Conceptual

Un mapa conceptual puede definirse como una herramienta de estudio que facilita el aprendizaje mediante el desarrollo de ideas de forma gráfica y esquematizada, vinculando distintos conceptos que, unidos en un todo congruente, ayudan a explicar un determinado tema a partir de los contenidos más relevantes del mismo.

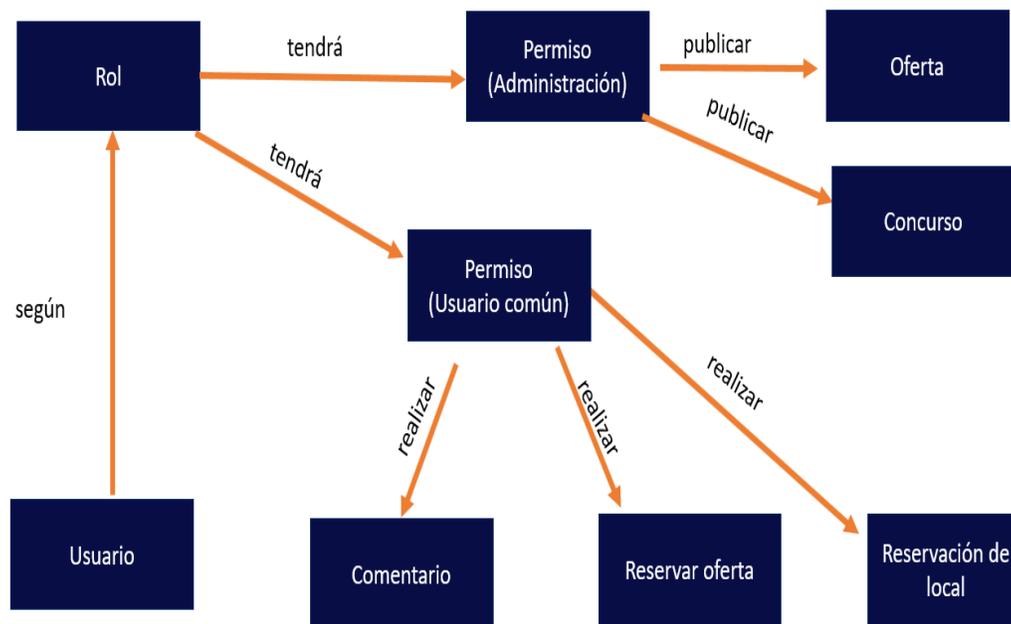


Figura 1: Mapa conceptual (Fuente: Elaboración propia)

2.3. Diseño

En esta fase se muestra los requisitos funcionales y no funcionales con lo que debe cumplir la propuesta de solución. Así como los artefactos generados a partir de la metodología seleccionada para guiar el proceso de desarrollo.

CAPÍTULO 2

2.3.1. Especificación de requisitos

De acuerdo con lo planteado en Sommerville (2011), en la ingeniería del software, los requisitos se utilizan como datos de entrada en la etapa de diseño del producto y establecen qué debe hacer el sistema, pero no cómo hacerlo. Son una condición o capacidad que un usuario necesita para poder resolver un problema o lograr un objetivo. De manera general estos requisitos son lo que el sistema debe hacer o una cualidad que el sistema debe poseer.

Requisitos funcionales

Según Sommerville (2011), los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. Luego de haber realizado un encuentro con el cliente se obtuvo un total de treinta y siete (37) requisitos funcionales, los cuales se relacionan a continuación en la Tabla 1. La prioridad de cada requisito se establece en función de la importancia fijada por el cliente a partir de sus necesidades.

Tabla 1: Listado de requisitos funcionales

| Código | Requisito funcional | Prioridad |
|---------------|-----------------------------|------------------|
| RF1 | Registrar usuario | Media |
| RF2 | Autenticar usuario | Media |
| RF3 | Modificar usuario | Media |
| RF4 | Eliminar usuario | Media |
| RF5 | Listar usuario | Media |
| RF6 | Mostrar detalles de usuario | Media |
| RF7 | Crear publicación | Alta |
| RF8 | Modificar publicación | Alta |
| RF9 | Eliminar publicación | Baja |
| RF10 | Listar publicación | Alta |
| RF11 | Publicar entrada | Alta |
| RF12 | Crear comentario | Media |

CAPÍTULO 2

| | | |
|-------------|---|-------|
| RF13 | Modificar comentario | Media |
| RF14 | Eliminar comentario | Baja |
| RF15 | Listar comentario | Media |
| RF16 | Publicar comentario | Media |
| RF17 | Votar por una publicación | Baja |
| RF18 | Crear concurso | Alta |
| RF19 | Modificar concurso | Alta |
| RF20 | Eliminar concurso | Baja |
| RF21 | Listar concurso | Alta |
| RF22 | Publicar concurso | Alta |
| RF23 | Crear oferta | Alta |
| RF24 | Modificar oferta | Alta |
| RF25 | Eliminar oferta | Baja |
| RF26 | Listar oferta | Alta |
| RF27 | Publicar oferta | Alta |
| RF28 | Mostrar detalles de oferta | Alta |
| RF29 | Reservar oferta | Alta |
| RF30 | Listar reservaciones | Alta |
| RF31 | Crear reserva de actividades en los teatros de la Universidad | Alta |
| RF32 | Modificar reserva de actividades en los teatros de la Universidad | Alta |
| RF33 | Eliminar reserva de actividades en los teatros de la Universidad | Baja |
| RF34 | Listar reserva de actividades en los teatros de la Universidad | Alta |
| RF35 | Mostrar detalles de reserva de actividades en los teatros de la Universidad | Alta |
| RF36 | Aprobar reserva de actividades en los teatros de la Universidad | Alta |
| RF37 | Exportar planificación por fecha, mes | Alta |

CAPÍTULO 2

Requisitos no funcionales

Distribuidos, según el estándar de calidad internacional ISO/IEC 9126-1, en especificaciones de usabilidad, confiabilidad, portabilidad, eficiencia, funcionalidad y mantenibilidad, se obtuvo un total de trece (13) requisitos no funcionales y quedan relacionados a continuación.

Usabilidad:

RnF1: El Sistema para la gestión procesos de la Dirección de Extensión Universitaria deberá ser una aplicación web

Confiabilidad:

RnF2: El sistema debe ser tolerante a fallos, y mostrar solo la información necesaria para orientar al usuario

Portabilidad:

RnF3: La propuesta de desarrollo debe ser capaz de ejecutarse en los navegadores empleados en la UCI (por ejemplo: Opera, Firefox, Chrome), así como adaptar su interfaz a cualquier dispositivo

Eficiencia:

RnF4: El sistema debe permitir que los usuarios (500) interactúen con él de manera concurrente

RnF5: El tiempo de demora de una petición al servidor debe ser menor a cinco (5) segundos

Funcionalidad:

RnF6: La aplicación debe gestionar y requerir información de usuarios para su uso

RnF7: La información manejada por el sistema estará protegida de accesos no autorizados

RnF8: Ante los errores que puedan ocasionarse en el sistema no se deben mostrar detalles de información que puedan comprometer su seguridad e integridad

RnF9: El sistema deberá ser utilizado por los usuarios con dominio uci.cu

RnF10: El sistema deberá ser capaz, por cuestiones de seguridad, expirar la sesión del usuario una vez pasado diez (10) minutos de inactividad

CAPÍTULO 2

Mantenibilidad:

RnF11: Se debe hacer uso de los estándares de codificación definidos para el sistema de gestión de procesos

RnF12: Se permitirá realizar modificaciones posteriores para adaptar mejoras al sistema o en caso que cambien las necesidades de los clientes

RnF13: El software estará bien documentado de forma tal que el tiempo de mantenimiento sea mínimo en caso de necesitarse

2.3.2. Historias de usuario

De acuerdo a lo que plantea Sánchez (2015), la metodología *AUP-UCI*, en su escenario 4 para la disciplina Requisitos, genera como uno de sus artefactos, las Historias de Usuario (HU), que consiste en una técnica para encapsular los requisitos del *software*, a través de un conjunto de tablas en las cuales el cliente describe brevemente las características que debe poseer el sistema.

Para el diseño de la propuesta de solución fueron generadas un total de treinta y siete (37) Historias de Usuario, a continuación, se muestran las correspondientes a los **RF-25** y **RF-30**, detallados en el sub-epígrafe 2.3.1.

Tabla 2: Historia de usuario Crear oferta

| Historia de Usuario | |
|--|------------------------------|
| Número: HU_25 | Nombre: Crear oferta |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir crear una oferta teniendo en cuenta que hay que llenar los siguientes campos como son nombre, lugar, fecha y capacidad de dicha oferta. Además, se debe mostrar un mensaje de confirmación para crear la misma. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz: | |

CAPÍTULO 2

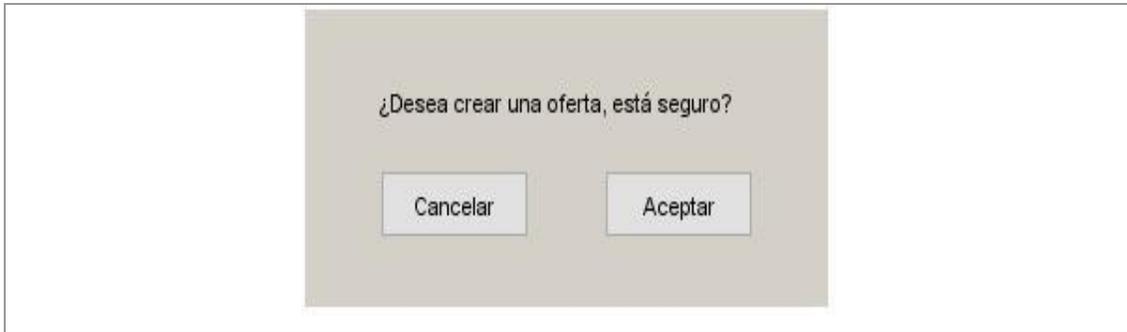
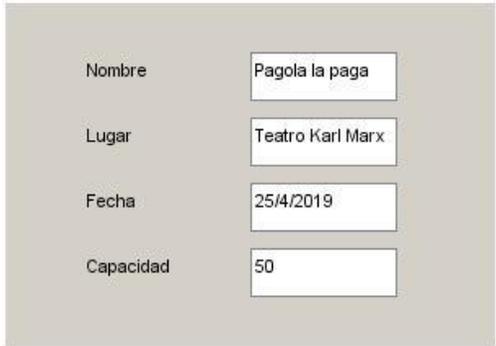


Tabla 3: Historia de usuario Detalles de oferta

| Historia de Usuario | |
|--|-----------------------|
| Número: HU_30 | Detalles de oferta |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir consultar los detalles de la oferta, como son el nombre, lugar, fecha y capacidad de la misma. Además, se debe mostrar un mensaje de confirmación. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz:  | |

2.3.3. Estilo arquitectónico

Modelo – Plantilla – Vista

El *framework* de desarrollo web *Django* emplea una modificación del estilo arquitectónico Modelo-Vista-Controlador (MVC), llamada *Model-Template-View* (MTV), que sería Modelo-Plantilla-Vista, esta forma de trabajar permite que sea pragmático.

Como plantea Infante-Montero (2012), para comprender como funciona el MTV de *Django* se debe tener en cuenta, su analogía con el MVC de la siguiente manera:

CAPÍTULO 2

- El modelo en *Django* continúa siendo Modelo
- La vista en *Django* pasa a llamarse Plantilla
- El controlador en *Django* pasa a llamarse Vista

A partir de lo planteado se asumirá el estilo arquitectónico MTV de *Django* del cual se explica su funcionamiento a continuación y se podrá observar en la Figura 2:

1. El navegador web envía una solicitud
2. La vista interactúa con el modelo para obtener los datos
3. La vista llama a la plantilla
4. La plantilla muestra la respuesta a la solicitud del navegador

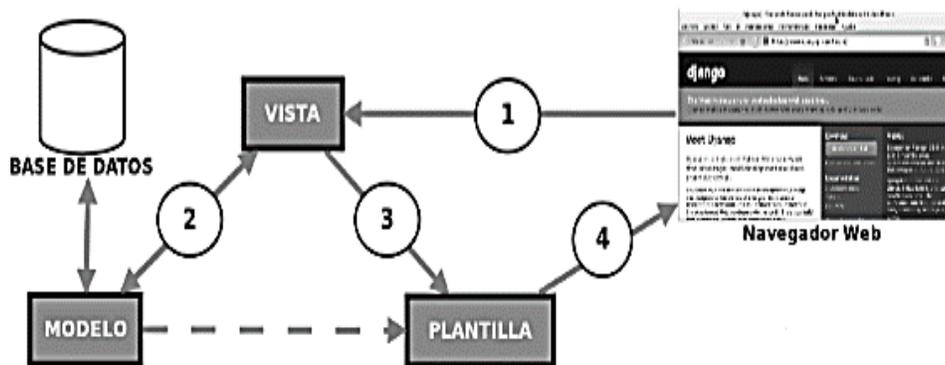


Figura 2: Funcionamiento del MTV de Django (Infante-Montero, 2012)

Del funcionamiento del MTV de *Django* se deriva lo siguiente (Infante-Montero, 2012):

- **El modelo:** Define los datos almacenados, es representado en forma de clases de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros igualmente posee métodos. Todo esto permite indicar y controlar el comportamiento de los datos.
- **La vista:** Su propósito es determinar qué datos serán visualizados, puede estar representado en forma de funciones o también en clases de *Python*. El *ObjectRelationalMapping* (*ORM*, es español Mapeo Relacional de Objetos) de *Django* permite escribir código *Python* en lugar *SQL* para hacer las consultas. Además, se encarga de tareas como: el envío de correo electrónico, autenticación con servicios externos, así como la validación de datos a través de formularios.
- **La plantilla:** Es la encargada de recibir los datos que provienen de las vistas y luego

CAPÍTULO 2

organizarlos para la presentación al navegador web. Es básicamente una página *HTML* con algunas etiquetas extras que son propias de *Django* que permiten que sea más flexible para los desarrolladores del *frontend*⁵.

2.3.4. Diagrama de clases del diseño

Los diagramas de clase (DC) pueden usarse cuando se desarrolla un modelo de sistema orientado a objetos para mostrar las clases en un sistema y las asociaciones entre dichas clases (Sommerville, 2011).

Los diagramas de clases del diseño con estereotipos web, describen gráficamente las especificaciones del modelo, la vista y la plantilla de las historias de usuario descritas en el sub-epígrafe 2.3.2. Estas representaciones contienen información acerca de las clases, asociaciones, atributos, métodos y dependencias.

Para el diseño de la propuesta de solución fueron generadas un total de treinta y siete (37) DC, a continuación, se muestran dos (2) de estos correspondientes a los RF-5 y RF-16 respectivamente, detallados en el sub-epígrafe 2.3.1.

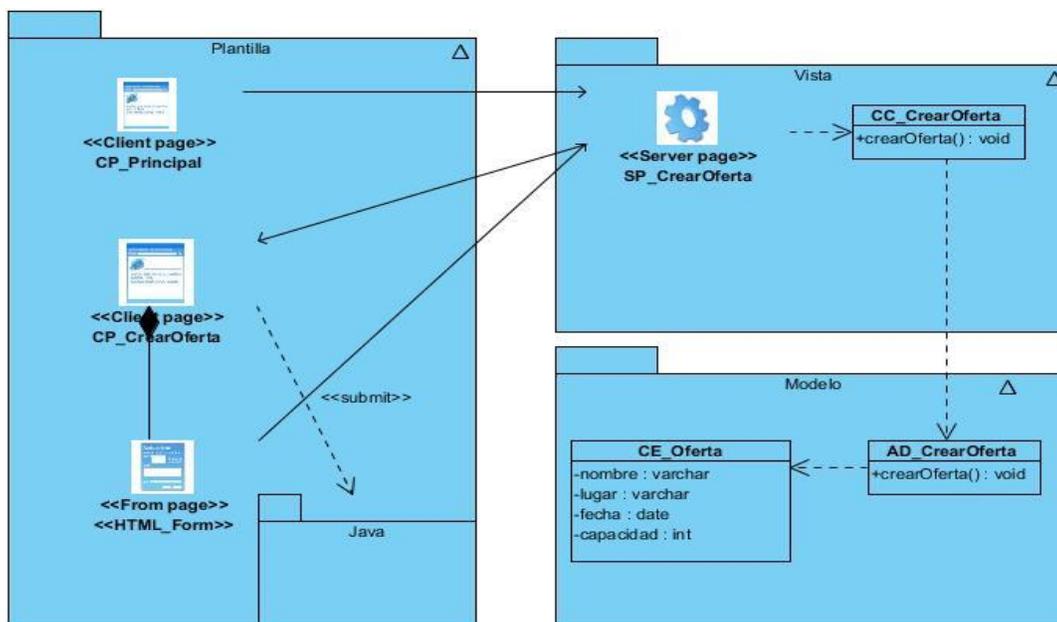


Figura 3: Diagrama de clase Crear oferta (Fuente: Elaboración propia)

⁵Es la parte del desarrollo web que se dedica de la parte frontal de un sitio web, en pocas palabras del diseño de un sitio web, desde la estructura del sitio hasta los estilos como colores, fondos, tamaños hasta llegar a las animaciones y efectos.

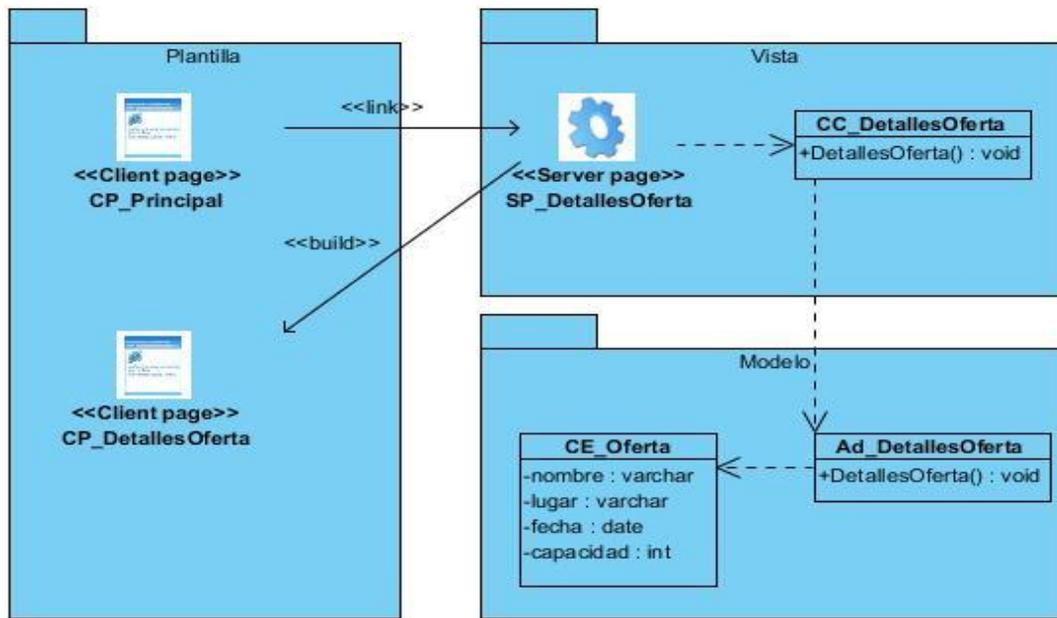


Figura 4: Diagrama de clase Detalles de oferta (Fuente: Elaboración propia)

2.3.5. Patrones de diseño

De acuerdo a las valoraciones de (Larman, 2004) los patrones de diseño representan la descripción de un problema particular y recurrente, que aparece en contextos específicos, y presenta un esquema genérico demostrado con éxito para su solución; este último se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos, así como también la forma como estos colaboran entre sí.

Seguidamente se presentan los patrones de diseño empleados en el sistema para la gestión de procesos en la dirección de extensión universitaria.

Patrones Generales de Software para la Asignación de Responsabilidades (GRASP):

Los patrones *GRASP* describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

GRASP es un acrónimo que significa *General Responsibility Assignment Software Patterns* (patrones generales de software para asignar responsabilidades). El nombre se eligió para indicar la importancia de captar (*grasping*) estos principios, si se quiere diseñar eficazmente el *software* orientado a objetos (Gamma, Helm, & R. Vlissides, 2009).

Según plantea (Astudillo, Visconti, & Hernán, 2011) lo patrones *GRASP* son:

- **Experto:** El patrón garantiza que la responsabilidad de la creación de un objeto o

CAPÍTULO 2

la implementación de un método, recaiga sobre la clase que conoce toda la información necesaria para crearlo lo que contribuye a un adecuado encapsulamiento, favoreciendo la robustez y fácil mantenimiento del sistema.

La clase entidad **db_offer**, es experta en la información relacionada con las ofertas que se estén brindando en ese momento.

- **Creador:** Se asigna la responsabilidad a una clase de crear cuando contiene, agrega, compone, almacena o usa otra clase, lo que brinda una alta posibilidad de reutilizar la clase creadora.
- **Bajo acoplamiento:** Es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras clases.

Este patrón ya viene incluido con *Django* que permite un bajo acoplamiento entre las piezas, lo que evita las dependencias, por ejemplo, a la hora de realizar cambios en las configuraciones de las *URL*⁶, en la BD y las plantillas *HTML*, basta solo con realizarlo una sola vez.

- **Alta Cohesión:** Asigna responsabilidades de manera que una clase no tenga muchas funcionalidades no relacionadas o no realice un trabajo excesivo. Este patrón incrementa la claridad y facilita la comprensión del diseño.

Una de las características de *Django* es la organización del trabajo en cuanto a la estructura del proyecto, lo cual permite crear y trabajar con clases con una alta cohesión. Por ejemplo, se puede observar en el sistema que cada clase controladora se ajusta a manejar solo las responsabilidades correspondientes a las entidades con las que se relaciona. Esto hace posible que el sistema sea flexible a cambios sustanciales con efecto mínimo.

- **Controlador:** Asignar la responsabilidad de administrar un mensaje de evento del sistema a una clase que representa el sistema global, dispositivo, subsistema o representa un escenario de caso de uso en el que tiene lugar el evento del sistema.

Este patrón es empleado en todo el sistema debido a que cada uno de los eventos

⁶Del inglés *UniformResourceLocator* (Localizador De Recursos Uniforme).

generados por el usuario es redirigido a una clase o función controladora que realiza las operaciones solicitadas, manteniendo siempre la alta cohesión.

Patrones Gang of Four (GoF):

El catálogo de patrones más famoso es el contenido en el libro “*Design Patterns: Elements of Reusable Object-Oriented Software*”, también conocido como: El libro *GOF (Gang-Of-Four Book)*. Según este documento, estos patrones se clasifican según su propósito en creacionales, estructurales y de comportamiento, mientras que respecto a su ámbito se clasifican en clases y objetos:

- **Decorator (Decorador):** Permite agregar responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades.

En muchos lenguajes de programación, el patrón decorador se implementa a través de subclases (herencia). En *Python*, se puede emplear la función de decorador incorporado. Un decorador de *Python* es un cambio específico a la sintaxis de este lenguaje, que se utiliza para extender el comportamiento de una clase, método o función sin utilizar la herencia (Kasampalis, 2015).

2.3.6. Modelo de datos

Un modelo de datos es una serie de conceptos que puede utilizarse para describir un conjunto de datos y las operaciones para manipularlos. En el diseño de bases de datos se usan primero los modelos conceptuales para lograr una descripción de alto nivel de la realidad, y luego se transforma el esquema conceptual en un esquema lógico. El motivo de realizar estas dos etapas es la dificultad de abstraer la estructura de una base de datos que presente cierta complejidad. Un esquema es un conjunto de representaciones lingüísticas o gráficas que describen la estructura de los datos de interés (Cuaran & Alonso-Guerrero, 2009).

El modelo de datos generado, como se observa en la Figura 5, está compuesto por dieciséis (16) relaciones y doce (12) tablas, las cuales se explican a continuación, para un mejor entendimiento del modelo:

Auth_user: registra, de un usuario del sistema, el atributo que lo identifica (*id*), su usuario *uci*, la contraseña, el correo, su(s) nombre(s) y apellidos, la fecha en que accedió por primera vez

CAPÍTULO 2

al sistema, así como la fecha de su última conexión.

Db_comment: contiene los comentarios que son realizados en las publicaciones en general, referentes a las ofertas publicadas. Contiene además las referencias al usuario que realiza el comentario (*id*).

Db_contest: contiene toda la información referente al concurso, así como su descripción, nombre, fecha de inicio del concurso y su culminación.

Db_postentry: abarca todo lo referente a las publicaciones como son el título, fecha de publicación, autor que lo identifica (*create_by_id*).

Db_reservationlocal: contiene todo lo referido a la reservación de los locales como es fecha de la actividad, tipo de actividad, si está disponible o no el local (*status*).

Auth_group: almacena la información de cada rol que se le ha asignado a cada usuario como son el nombre (*name*) y el identificador (*id*).

Db_usserprofile: incluye la información de la entidad usuario ya que hereda de la misma, además de tener un identificador propio (*id*).

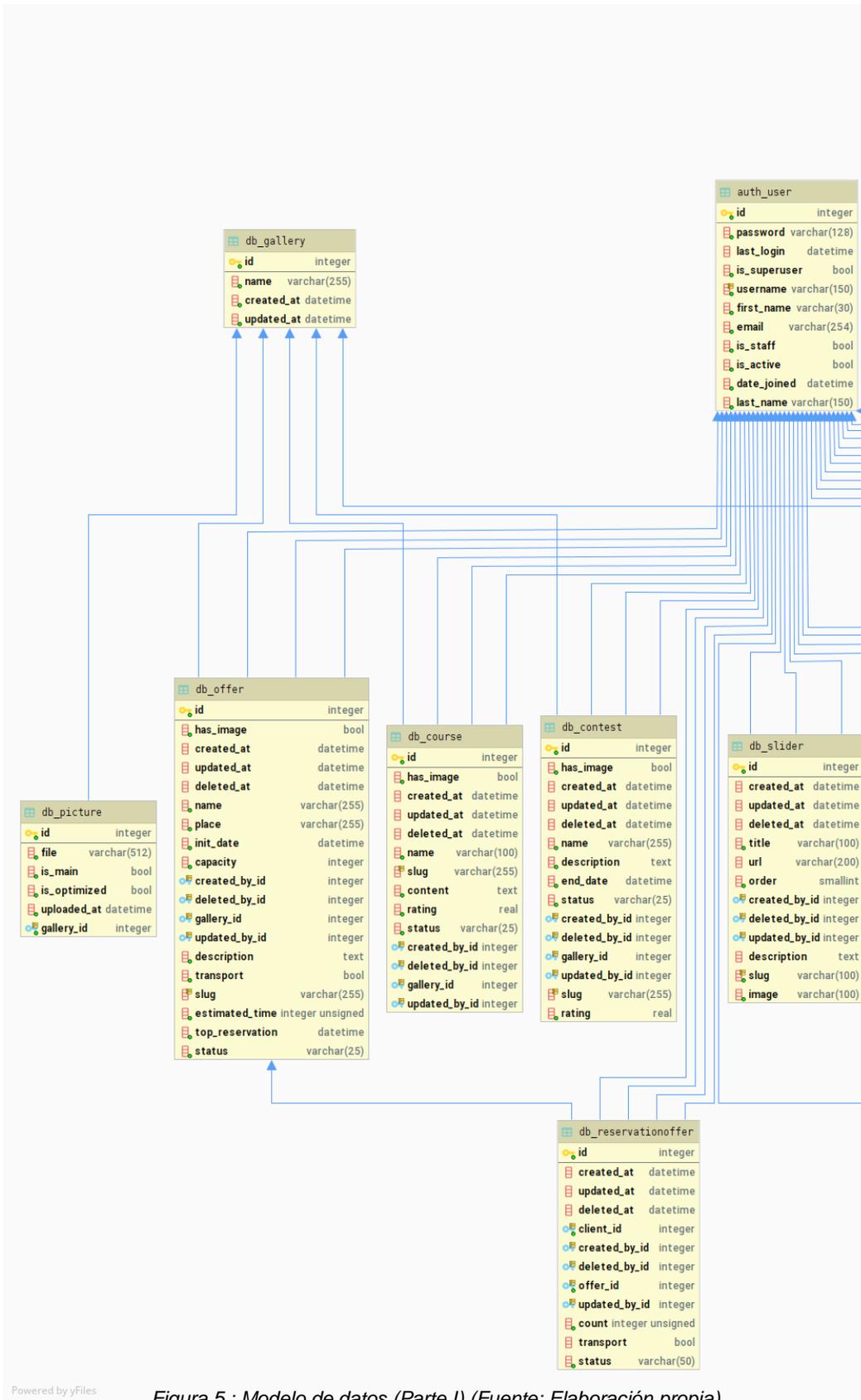
Db_nomvalue: almacena todos los nomencladores, así como su tipo (*type_id*) y su nombre (*name*) además de poseer un identificador (*id*) para cada uno.

Auth_permission: contiene todos los permisos (*id*) que le son concedidos a cada usuario.

Db_reservationoffer: contiene todo lo referente a la reservación de la oferta propuesta como es su estado (*status*) y su identificador (*id*).

Db_offer: almacena la información de las ofertas que son creados en el sistema, su identificador (*id*), la capacidad, así como el lugar.

CAPÍTULO 2



Powered by yFiles

Figura 5 : Modelo de datos (Parte I) (Fuente: Elaboración propia)

CAPÍTULO 2

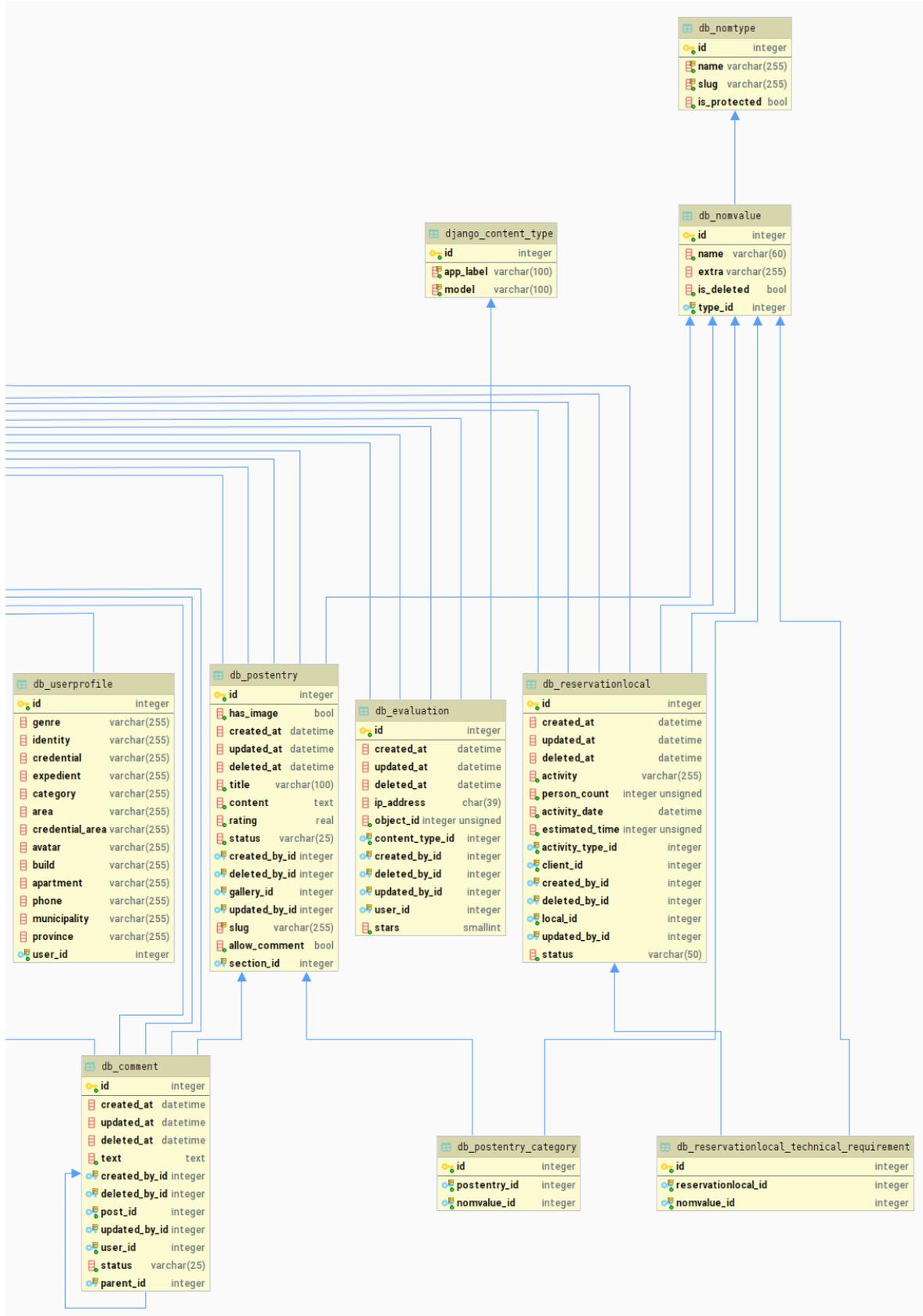


Figura 6 Modelo de datos (Parte II) (Fuente: Elaboración propia)

CAPÍTULO 2

2.3.7. Modelo de despliegue

Un modelo de despliegue refleja la arquitectura en tiempo de ejecución de un sistema. En la Figura 6 se representan los dispositivos físicos de los artefactos de software en nodos como Servidor de Aplicaciones (*Apache*), Sistema Gestor de Base de Datos (*PostgreSQL*), Pasarela de autenticación del dominio mediante *WebServices* y el ordenador cliente, además especifica el protocolo y puerto que se emplea en la conexión entre cada nodo.

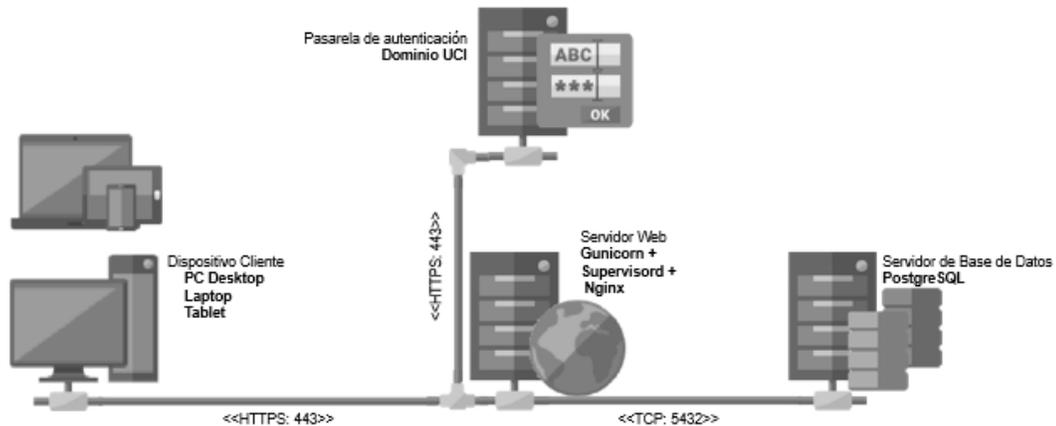


Figura 7: Modelo de despliegue de la solución propuesta (Fuente: Elaboración propia)

Este diagrama se considera para lograr un despliegue exitoso de la aplicación. En este se definen:

Dispositivo Cliente: Se refiere a las estaciones de trabajo (Cliente: PC, tableta, teléfono celular) que el usuario utilizará para conectarse, vía *HTTPS*, con el servidor de aplicaciones web.

Servidor de Bases de Datos: Almacena toda la información que brinda el sistema hospedado en el servidor de aplicaciones. La información es obtenida o modificada en dependencia del nivel de privilegio del usuario que realiza la petición. La comunicación con el servidor de aplicaciones es a través del protocolo *TCP* empleando el puerto 5432. Además, debe contar con una capacidad mínima de 500 MB de disco duro.

Pasarela Autenticación: Es la encargada de brindar la información de los usuarios de la UCI que se registran en el sistema hospedado en el servidor de aplicaciones. La comunicación la realiza a través del protocolo seguro *HTTPS*.

Servidor Web: Es el encargado de brindar la interfaz del sistema para que los usuarios

CAPÍTULO 2

puedan hacer uso de este, almacena todo el código fuente del sistema y se comunica por medio de los protocolos *TCP* con el servidor de bases de datos y por *HTTPS* con la pasarela de autenticación para usuarios de la *UCI*. Este debe disponer de un ordenador con sistema operativo *GNU/Linux* y cuyas propiedades mínimas sean un procesador *Intel Dual Core* con velocidad de 2.10 GHz, 2 GB de memoria *RAM*. El cual funciona de la siguiente manera: Nginx, inicialmente, es el encargado de resolver la petición web realizada por el cliente, reenviándola para que Gunicorn pueda procesarla utilizando varios hilos de ejecución. Cada uno de estos hilos es capaz de comunicarse con la aplicación Django de manera que pueden atenderse múltiples peticiones al mismo tiempo, manteniendo un buen tiempo de respuesta. Por otro lado, Supervisor se encarga de controlar la ejecución del proceso Gunicorn de tal forma que podamos iniciar, detener o reiniciar la aplicación de Django en el momento que sea necesario o de ocurrir cualquier error.

2.4. Conclusiones del capítulo

Luego de haber de realizar el análisis y diseño del módulo de implementación de sentencias de manipulación y haber generado los artefactos que dispone la metodología *AUP en la variante UCI escenario 4*, se puede concluir lo siguiente:

- El análisis de las características del sistema y el modelado del dominio, permitió identificar los principales requisitos funcionales y no funcionales del sistema para la gestión de procesos de la Dirección de Extensión Universitaria los que fueron agrupados y categorizados en sus respectivas historias de usuarios.
- La identificación de los patrones de diseño y el estilo arquitectónico de la solución propuesta, permite disminuir el impacto de los cambios futuros en el código fuente de la misma.
- El diseño de los diagramas de clases, facilitó el enfoque en cuanto a composición lógica y física de la propuesta de solución.
- La generación de todos los artefactos requeridos por el modelo de desarrollo, documentan la solución propuesta, lo cual facilita su posterior mantenimiento (actualización o adición de funcionalidades).

CAPÍTULO 3: Implementación y pruebas del Sistema para la gestión de procesos en la Dirección de Extensión Universitaria

3.1. Introducción

En el capítulo se aborda las actividades que se llevan a cabo durante la fase de implementación y pruebas. Se define un estándar de codificación para facilitar la comprensión del código. Además, se diseña y aplican las pruebas para comprobar el correcto funcionamiento del sistema. Igualmente se mostrarán las principales interfaces del mismo.

3.2. Estándar de codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del *software* y para obtener un buen rendimiento (Microsoft, 2017).

A continuación, se define el estándar de codificación empleado en el Sistema de gestión de procesos de la Dirección de Extensión Universitaria.

Tabla 4. Estándares de codificación a utilizar en la implementación del sistema.

| Tipo de estándar | Descripción |
|--------------------------------------|--|
| Organización del código | <ul style="list-style-type: none">➤ El código en una página se organizará por bloques➤ El formato del código se ajustará por la opción <i>ReformatCode</i> del <i>IDE</i> de desarrollo <i>PyCharm</i>➤ La indentación se realizará solamente con tabulaciones, no debe utilizarse nunca los cuatro (4) espacios |
| Máxima longitud de las líneas | <ul style="list-style-type: none">➤ Todas las líneas deben estar limitadas a un máximo de setenta y nueve (175) caracteres |

CAPÍTULO 3

| | |
|---|--|
| | <ul style="list-style-type: none">➤ Dentro de paréntesis, corchetes o llaves se puede utilizar la continuación implícita para cortar las líneas largas➤ En cualquier circunstancia se puede utilizar el carácter “\” para cortar las líneas largas |
| Líneas en blanco | <ul style="list-style-type: none">➤ Separar las funciones de alto nivel y definiciones de clases con dos (2) líneas en blanco➤ Las definiciones de métodos dentro de una clase deben separarse por una (1) línea en blanco➤ Se puede utilizar líneas en blanco escasamente para separar secciones lógicas |
| Codificaciones | <ul style="list-style-type: none">➤ Utilizar la codificación <i>UTF-8</i>➤ Se pueden incluir cadenas que no correspondan a esta codificación utilizando “\x”, “\u” o “\U” |
| Importación | <ul style="list-style-type: none">➤ Las importaciones deben estar en líneas separadas➤ Siempre deben colocarse al comienzo del archivo➤ Deben quedar agrupadas de la siguiente forma:<ul style="list-style-type: none">○ Importaciones de la librería estándar○ Importaciones terceras relacionadas○ Importaciones locales de la aplicación / librerías➤ Cada grupo de importaciones debe estar separado por una línea en blanco |
| Espacios en blanco en expresiones y sentencias | <ul style="list-style-type: none">➤ Evitar utilizar espacios en blanco en las siguientes situaciones:➤ Inmediatamente dentro de paréntesis, corchetes y llaves➤ Inmediatamente antes de una coma, un punto y coma o dos puntos➤ Inmediatamente antes del paréntesis que comienza la lista de argumentos en la llamada a una función➤ Inmediatamente antes de un corchete que empieza una indexación➤ Más de un espacio alrededor de un operador de asignación (u otro) para alinearlos con otro➤ Deben rodearse con exactamente un espacio los siguientes operadores binarios: |

CAPÍTULO 3

| | |
|-------------------------------------|---|
| | <ul style="list-style-type: none">➤ Asignación (=)➤ Asignación de aumentación (+=, -=, *=, /=)➤ Comparación (==, <, >, >=, <=, =, <>, <i>in</i>, <i>not in</i>, <i>is</i>, <i>is not</i>)➤ Expresiones lógicas (<i>and</i>, <i>or</i>, <i>not</i>)➤ Si se utilizan operadores con prioridad diferente se aconseja rodear con espacios a los operadores de menor prioridad➤ No utilizar espacios alrededor del igual (=) cuando es utilizado para indicar un argumento de una función o un parámetro con un valor por defecto |
| Comentarios | <ul style="list-style-type: none">➤ Los comentarios deben ser oraciones completas➤ Si un comentario es una frase u oración su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula➤ Si un comentario es corto el punto final puede omitirse➤ Los comentarios de una línea para aclaraciones del código aparecerán seguidos de los caracteres “//” en caso de código <i>JavaScript</i> mientras que en <i>Python</i> por el carácter “#” y deben ubicarse en la misma línea que se desea comentar➤ Los comentarios de varias líneas para organización del código aparecerán dentro de los caracteres “/** ... */” en caso de código <i>JavaScript</i>, mientras que en <i>Python</i> se hará con los caracteres “...” |
| Convenciones de nombramiento | <ul style="list-style-type: none">➤ Nunca se deben utilizar como simple caracteres para nombres de variables los caracteres en minúscula “l”, o mayúscula “O”, o la mayúscula “L” ya que en algunas fuentes son indistinguibles de los números uno (1) y cero (0)➤ Los módulos deben tener un nombre corto y en minúscula.➤ Los nombres de clases deben utilizar la convención “<i>CapWords</i>” (palabras que comienzan con mayúsculas)➤ Los nombres de las funciones deben estar escrito en minúscula separando las palabras con un guión bajo “_”➤ Las constantes deben quedar escritas con letras mayúsculas |

CAPÍTULO 3

| | |
|--|--|
| | separando las palabras por un guión bajo (_) |
|--|--|

3.3. Pruebas de Software

El proceso de pruebas se centra en los procesos lógicos internos del *software*, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las pruebas para la detección de errores. Además, son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa (Pressman, 2010).

En este epígrafe se muestran los resultados de la estrategia de prueba diseñada para la propuesta de solución (ver Tabla 5), en función de garantizar y validar la calidad de este.

Tabla 5. Estrategia de pruebas.

| Tipo de prueba | Método (técnica) de prueba | Validación |
|-----------------------|-------------------------------|---|
| Funcional | Casos de prueba (Caja Negra) | Valida las funcionalidades diseñadas para el sistema |
| Carga y estrés | <i>Software Apache JMeter</i> | Valida el comportamiento del sistema con distintos niveles de usuarios concurrentes y el consumo excesivo de sus recursos |
| Seguridad | <i>Software Acunetix</i> | Valida la confidencialidad, integridad y disponibilidad de los datos en el sistema |
| Usabilidad | Lista de Chequeo | Valida, a partir de sus características, la capacidad del sistema de cumplir con el propósito para el que fue diseñado |

3.3.1. 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 estas pruebas se realiza con la intención de detectar funciones incorrectas o ausentes, errores en accesos a bases de datos externas, errores de interfaz,

CAPÍTULO 3

errores de rendimiento, y errores de inicialización y de terminación. Dentro de la prueba se incluyen la técnica de partición de equivalencia que será la empleada en la validación. Esta técnica divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba (Pressman, 2010).

Se realizaron los diseños de casos de prueba para validar la propuesta de solución implementada para comprobar que la herramienta agregue correctamente una oferta al sistema.

A continuación, en la Tabla 6, se muestran las variables empleadas en el caso de prueba para el escenario “Crear Oferta”.

Tabla 6: Variables empleadas en el Caso de Prueba del RF25 Crear oferta

| No. | Nombre del campo | Clasificación | Nulo | Descripción |
|-----|------------------|-------------------------|------|---|
| 1 | Nombre | Campo de Texto | NO | Contiene solo caracteres alfanuméricos. |
| 2 | Descripción | Área de Texto | NO | Contiene todo tipo de carácter. |
| 3 | Capacidad | Campo de numérico | NO | Contiene solo números. |
| 4 | Fecha | Campo de fecha | NO | Contiene formato de fecha (día/mes/año hora: minutos) |
| 5 | Lugar | Campo de Texto | NO | Contiene solo letras y números. |
| 6 | Tiene Imagen | Casilla de verificación | SI | Valores Verdadero o Falso. |
| 7 | Transporte | Casilla de verificación | Si | Valores Verdadero o Falso. |

Las celdas de la tabla contienen V, I, o N/A. V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

CAPÍTULO 3

Tabla 7: Caso de Prueba del RF25 Crear oferta.

| Escenario | Descripción | V1 | V2 | V3 | V4 | V5 | V6 | V7 | Respuesta | Flujo central |
|--|--|-----------------------------------|-------------------|---------|-------------------|--------------------|----|----|--|--|
| EC. 1.1 Crear oferta. Datos correctos. | Se registran los datos de la oferta correctamente. | V | V | V | V | V | V | V | Muestra el mensaje "Se ha creado la oferta (nombre de la oferta)" | El gestor de ofertas rellena los campos de forma correcta y da clic en el botón Guardar. |
| | | Show humorístico "Pagola la paga" | Descripción (...) | 50 | 12/4/2019 9:00 PM | Teatro Karl Marx | NO | NO | | |
| EC. 1.2 Crear oferta. Datos incorrectos. | Se registran los datos de la oferta incorrectamente. | V | V | V | V | I | V | V | Muestra el mensaje "El campo Lugar solo contiene caracteres alfanuméricos" | El gestor de ofertas rellena los campos, pero escribe un carácter extraño en un campo que solo contiene caracteres alfanuméricos y da clic en Guardar. |
| | | Show humorístico "Pagola la paga" | Descripción (...) | 50 | 12/4/2019 9:00 PM | Teatro Karl / Marx | NO | NO | | |
| EC. 1.2 Crear oferta. Campos vacíos. | Existen campos obligatorios vacíos. | I | V | I | V | I | V | V | Muestra el mensaje "Existen campos obligatorios vacíos (campos)" | El gestor de ofertas llena los campos, pero deja alguno vacíos y da clic en Guardar. |
| | | (Vacío) | Descripción (...) | (Vacío) | 12/4/2019 9:00 PM | (Vacío) | NO | NO | | |

CAPÍTULO 3

Resultados de las pruebas funcionales.

Las pruebas funcionales se realizaron en cuatro (4) iteraciones a medida que avanzaba el sistema, las cuales guiaron la calidad del mismo, y determinaron en cada momento si se estaba o no en condiciones de continuar avanzando con el ciclo de desarrollo.

Como resultado final de estas pruebas, se obtuvo, en una primera iteración, un total de dieciseis (16) no conformidades, divididas en cuatro (4) de ortografía, siete (7) de redacción, dos (2) de funcionalidad y tres (3) de validación. De estas, se resolvieron nueve (9), y siete (7) quedaron pendientes. En una segunda iteración, no se detectaron nuevas no conformidades y de las siete (7) pendientes, solo dos (2) se mantuvieron para la próxima iteración, donde fueron resueltas y no se detectan nuevas no conformidades. En una cuarta iteración no se identifican nuevas inconformidades, obteniendo, de esta manera, los resultados esperados. La gráfica de la Figura 7, muestra los resultados descritos anteriormente.

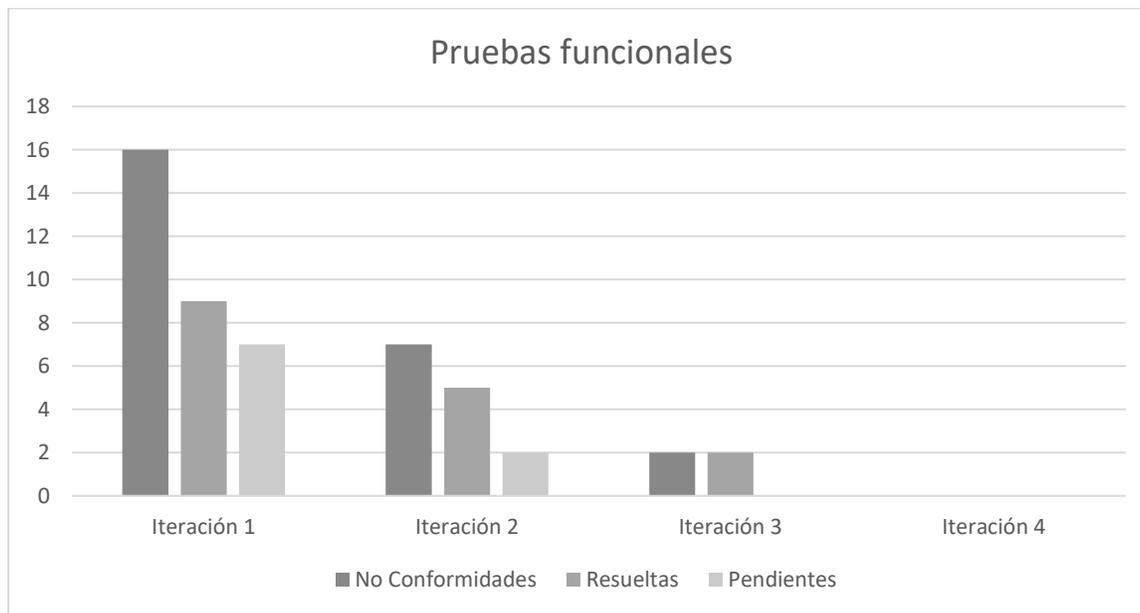


Figura 8: Resultados de las pruebas funcionales (Fuente: Elaboración propia)

Las no conformidades de funcionalidad, estuvieron relacionadas con crear una oferta y consultar las cátedras honoríficas. En el primer caso, el sistema no permitía crear la oferta sin que se seleccionara una imagen, sin embargo, el problema fue resuelto al permitir que el atributo image de la entidad Oferta, el cual estaba especificado como campo obligatorio, pudiese tomar valor nulo. En el último caso, se intentaba acceder a la sesión de cátedras

CAPÍTULO 3

honoríficas, pero dicho botón no accedía a la página correspondiente esto se debía a que la URL se encontraba mal configurada por lo que se corrigió.

Las no conformidades de validación fueron detectadas en el formulario para crear una oferta, donde este permitía introducir caracteres extraños en el nombre de una oferta. La solución para los caracteres extraños fue validar el nombre de una oferta mediante una expresión regular que permite solo el uso de caracteres alfanuméricos, las tildes, las diéresis, espacios, puntos y el guión bajo. Para el caso que se introduzca un caracter que no cumpla con la descripción de cada campo, el sistema responde con un mensaje en caso de no cumplir con dicha estructura.

3.3.2. Pruebas de carga y estrés

La prueba de carga y estrés se refiere, generalmente, a la práctica de comprobar el comportamiento de una aplicación mediante cargas o entradas pesadas. Las mismas se realizan con el fin de verificar si el sistema satisface los requisitos de rendimiento para situaciones críticas como pueden ser: la cantidad límite de usuarios accediendo de forma concurrente a los servicios brindados, documentos extremadamente grandes, cantidad de transacciones que se pueden procesar de forma concurrente cada minuto, tiempo de respuesta, entre otros (ITI, 2017).

Para realizar las pruebas de carga y estrés se efectuó un análisis de la cantidad de usuarios en la Universidad. Del cual se obtuvo que la misma cuenta con cerca de 8000 usuarios. Teniendo en cuenta que el sistema está pensado para el apoyo de la Dirección de Extensión Universitaria, se toma muestras para casos críticos en que se conecten entre 100 y 200 usuarios simultáneamente.

Las pruebas se desarrollaron con el apoyo de la herramienta *Apache JMeter 2.12*, en la que se simuló el entorno donde debe interactuar el sistema para obtener la información más correcta acerca del comportamiento y resultados en general. Por lo que fue elegido un ambiente con las siguientes características:

Hardware:

- Microprocesador: *Intel Core (TM) i3 – 3110M CPU @ 2.40 GHz*

CAPÍTULO 3

- Memoria RAM: 4GB
- Tarjeta de Red: *Ethernet* 10/100 Mbps

Software:

- Sistema Operativo: *GNU/Linux Ubuntu* 16.04 Arquitectura de 64bits

A continuación, se describen las variables que miden el resultado de las pruebas de carga y estrés realizadas al módulo:

Muestra: Cantidad de peticiones realizadas para cada *URL*.

Media: Tiempo promedio en milisegundos en el que se obtienen los resultados.

Mediana: Tiempo en milisegundos en el que se obtuvo el resultado que ocupa la posición central.

Min: Tiempo mínimo que demora un hilo en acceder a una página.

Max: Tiempo máximo que demora un hilo en acceder a una página.

Línea 90 %: Máximo tiempo utilizado por el 90 % de la muestra, al resto de la misma le llevo más tiempo.

% Error: Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.

Rendimiento (Rend): El rendimiento se mide en cantidad de solicitudes por segundo.

Kb/s: Velocidad de carga de las páginas.

Como se muestra en la Tabla 8, se simularon las peticiones realizadas al sistema por un total de 100, 500 y 1000 usuarios simultáneamente, realizando hasta 5 peticiones por segundo, obteniéndose los siguientes resultados:

Tabla 8: Resultado de las pruebas de carga y estrés

| Usuarios | Muestras | Media | Mediana | Min | Max | Línea 90% | % Error | Rend. | Kb/s |
|-------------|----------|-------|---------|-----|------|-----------|---------|-------|-------|
| 100 | 500 | 1642 | 1125 | 91 | 4571 | 2819 | 0.00% | 45.56 | 194.8 |
| 500 | 2500 | 1285 | 1078 | 125 | 4946 | 2096 | 0.45% | 216.3 | 542.3 |
| 1000 | 5000 | 1225 | 1023 | 69 | 4761 | 1703 | 2.89% | 437.7 | 974.6 |

3.3.3. Pruebas de seguridad

Las pruebas de seguridad permiten realizar una evaluación de los sistemas desde el punto de vista externo y sin conocimiento previo de este. Tienen como objetivo hacer un análisis con el fin de encontrar fallos de seguridad tanto en el diseño como en la implementación de la aplicación. Además, buscan medir la confidencialidad, integridad y disponibilidad de los datos, partiendo de la identificación de amenazas y riesgos en el uso de interfaces de usuarios final. Una vez terminadas las pruebas es posible medir y cuantificar los riesgos a los cuales se ven expuestos aplicativos de la infraestructura interna y externa.

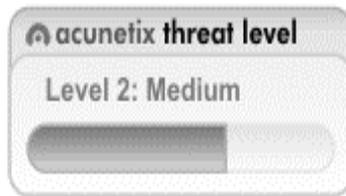
Resultados de las pruebas de seguridad.

Las pruebas de seguridad se aplicaron con ayuda de la herramienta *Acunetix Web Vulnerability Scanner 9.5* que establece alertas de tipo: alta, media, baja e informativa, realizándose en dos iteraciones del desarrollo de la propuesta solución

En una primera iteración se obtuvo un total de 15 alertas de seguridad, de las cuales 6 clasifican de nivel medio, 6 de nivel bajo y 3 informativas. De las de nivel medio, destacó el uso de protocolo no seguro para el envío de datos, así como los mensajes de error que se muestra en el modo DEBUG de *Django* para el desarrollo.

Para las de nivel bajo, se detectó problemas para la protección de contra ataques de fuerza bruta a la página de autenticación, así como directorios que pueden ser accesibles directamente sin pasar la autenticación y la protección de las cookies y las sesiones en el navegador. De carácter informativo fue detectadas una dirección de correo y una posible cuenta de usuario en un fichero.

CAPÍTULO 3



Acunetix Threat Level 2

One or more medium-severity type vulnerabilities have been discovered by the scanner. You should investigate each of these vulnerabilities to ensure they will not escalate to more severe problems.

| | |
|--------------------|----|
| Total alerts found | 15 |
| High | 0 |
| Medium | 6 |
| Low | 6 |
| Informational | 3 |

Figura 9: Pruebas de seguridad, primera iteración (Fuente: Acunetix WVS)

En la imagen de la Figura 9, se puede apreciar el resultado final luego de corregir todas las alertas detectadas por la herramienta.



Acunetix Threat Level 0

No vulnerabilities have been discovered by the scanner.

| | |
|--------------------|---|
| Total alerts found | 0 |
| High | 0 |
| Medium | 0 |
| Low | 0 |
| Informational | 0 |

Figura 10: Pruebas de seguridad. Resultado final de la segunda iteración (Fuente: Acunetix WVS)

3.3.4. Pruebas de usabilidad

En el contexto del desarrollo de software, la usabilidad está considerada como uno de los factores de calidad de mayor importancia para el éxito de un proyecto. De manera general, el término usabilidad es empleado para referirse a la capacidad que posee un producto de ser utilizado por los usuarios de forma fácil, eficiente y con satisfacción, en un determinado

CAPÍTULO 3

contexto de uso. La usabilidad se describe como un término multidimensional en el que intervienen cinco atributos fundamentales: capacidad de aprendizaje, eficiencia en el uso, facilidad de memorizar, tolerante a errores y subjetivamente satisfactorio (Nielsen, 2010).

La realización de pruebas de usabilidad contribuye, en cierta medida, a la adquisición de aplicaciones de alta calidad y gran facilidad de uso por parte de los usuarios finales (Alonso, Y. H.; Fortún, L. L., 2014). Para ello, especialistas del Departamento de Evaluación de Productos de Software han establecido una lista de chequeo que proporciona un conjunto de preguntas clasificadas en nueve (9) categorías.

Resultados de las pruebas de Usabilidad

A continuación, la Tabla 9 muestra los resultados de la aplicación de la lista de chequeo al Sistema para la gestión de procesos de la Dirección de Extensión Universitaria.

Tabla 9: Resultados de las pruebas de Usabilidad empleando lista de chequeo

| Categoría de indicadores | Indicadores | Evaluados | Correctos |
|---|--------------------|------------------|------------------|
| Visibilidad del sistema | 17 | 14 | 13 |
| Lenguaje común entre sistema y usuario | 11 | 7 | 7 |
| Libertad y control por parte del usuario | 29 | 14 | 14 |
| Consistencia y estándares | 33 | 22 | 21 |
| Estética y diseño minimalista | 18 | 9 | 9 |
| Prevención de errores | 8 | 6 | 6 |
| Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores | 11 | 7 | 1 |
| Ayuda y documentación | 11 | 4 | 2 |
| Flexibilidad y eficiencia de uso | 6 | 5 | 4 |
| TOTAL | 144 | 88 | 77 |

En la tabla anterior se puede apreciar que, de un total de 144 indicadores de usabilidad, fueron evaluados 88, de los cuales el Sistema para la gestión de procesos de la Dirección de Extensión Universitaria, cumple con 77 indicadores, cifra que representa el 87.5%. Para una mejor comprensión de los resultados obtenidos se han representado los mismos en la gráfica de la Figura 10.

CAPÍTULO 3

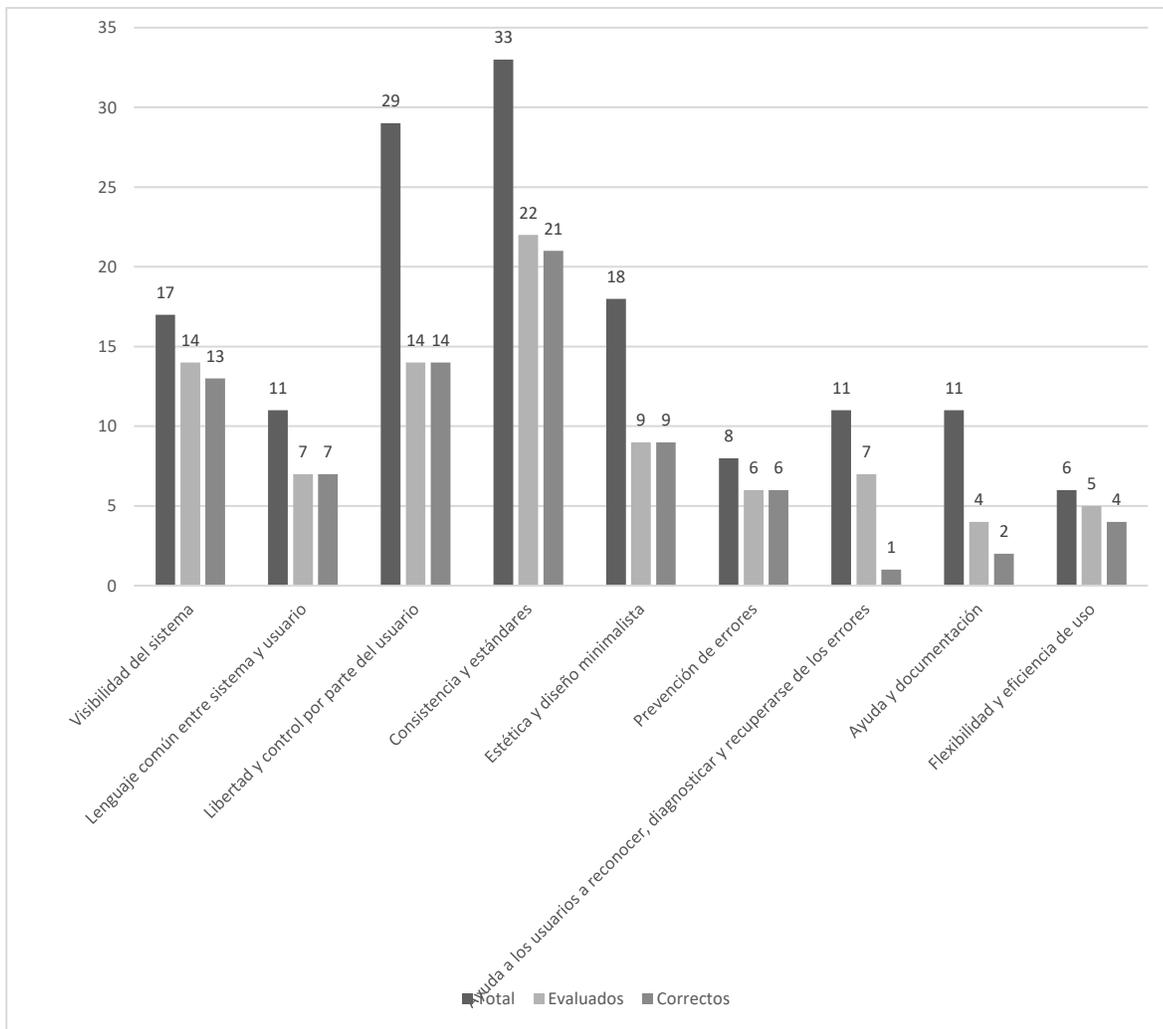


Figura 11: Cumplimiento de indicadores de la lista de chequeo de usabilidad (Fuente: Elaboración propia)

Después de analizar los resultados obtenidos en las pruebas de usabilidad, se identificaron nueve (9) indicadores con posibles mejoras de acuerdo al alcance del presente trabajo. Estos fueron implementados para facilitar el uso y la interacción entre el usuario y las operaciones que se realizan en el sistema; incrementando el nivel de usabilidad a 95.45%. En la gráfica de la Figura 11 se representa el estado del nivel de usabilidad resultante después de terminada la prueba.

CAPÍTULO 3

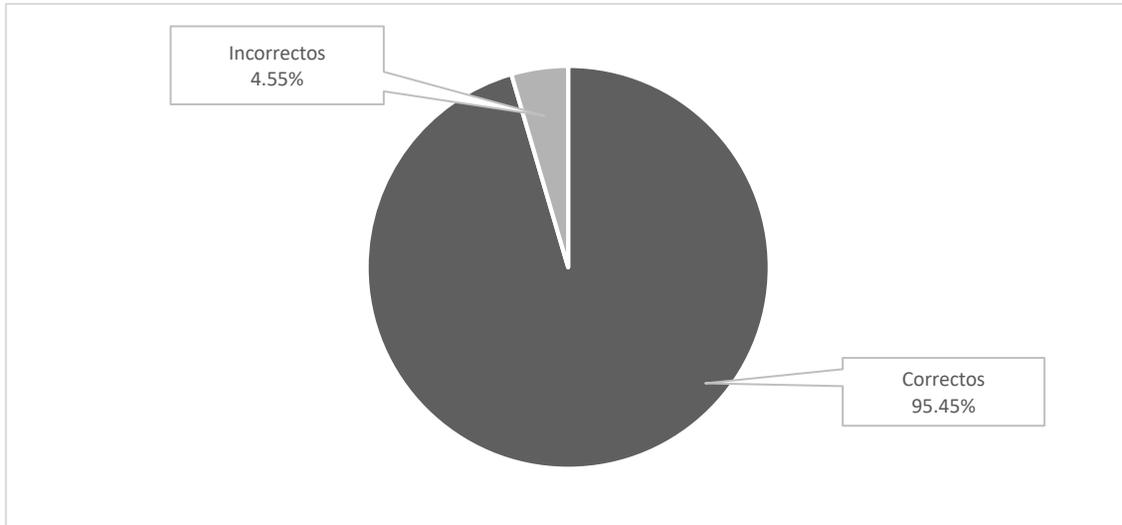


Figura 12: Nivel de usabilidad resultante (Fuente: Elaboración propia)

3.4. Interfaces principales

Una vez finalizado el desarrollo del software es posible visualizar las pantallas principales del Sistema para la gestión de Procesos ne la Dirección de Extensión Universitaria, donde se observa el resultado obtenido durante la implementación de las historias de usuarios descritas en el capítulo anterior.

CAPÍTULO 3

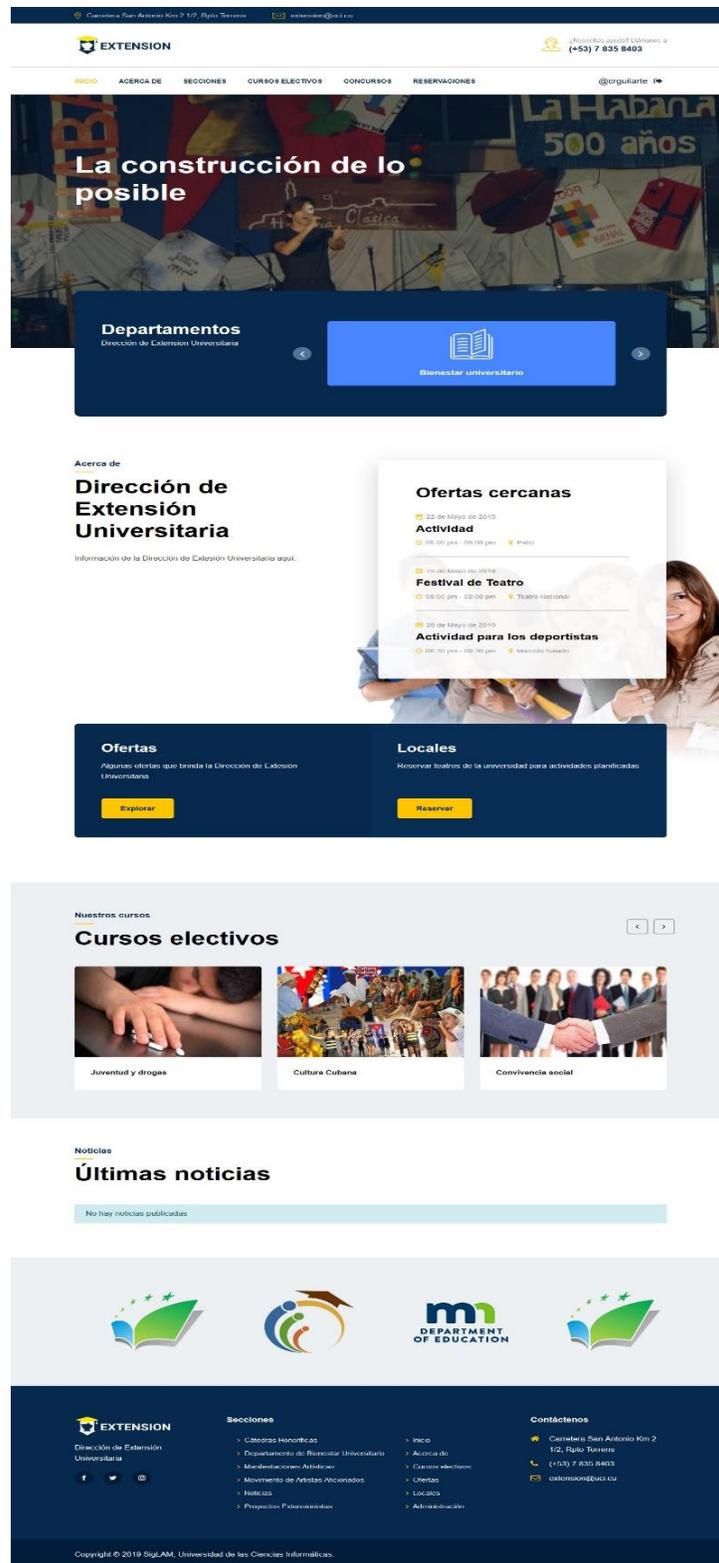


Figura 13: Interfaz principal. Portada

CAPÍTULO 3

Carretera San Antonio Km 2 1/2, Rpto Torrens extension@uci.cu

EXTENSION ¿Necesitas ayuda? Llámamos a (+53) 7 835 8403

INICIO ACERCA DE SECCIONES CURSOS ELECTIVOS CONCURSOS **RESERVACIONES** @crguilarte

Ofertas

Inicio / Oferta / **Actividad**

Actividad

22 de Mayo de 2019 08:00 pm - 09:00 pm Patio [Editar](#)

Se agotó el tiempo



Fiesta de los ...

- Entradas disponibles**
50
- Fecha**
22 de Mayo de 2019
- Hora inicio**
4 p.m.
- Hora fin**
5 p.m.
- Lugar**
Patio
- Transporte**
SI

EXTENSION
Dirección de Extensión Universitaria

Secciones

- > Cátedras Honoríficas
- > Departamento de Bienestar Universitario
- > Manifestaciones Artísticas
- > Movimiento de Artistas Aficionados
- > Noticias
- > Proyectos Extensionistas
- > Inicio
- > Acerca de
- > Cursos electivos
- > Ofertas
- > Locales
- > Administración

Contáctenos

- Carretera San Antonio Km 2 1/2, Rpto Torrens
- (+53) 7 835 8403
- extension@uci.cu

Copyright © 2019 SigLAM, Universidad de las Ciencias Informáticas.

Figura 14: Interfaz principal. Reservar oferta.

CAPÍTULO 3

Carretera San Antonio Km 2 1/2, Rpto. Torrens extension@uci.cu

 ¿Necesitas ayuda? Llámamos a (+53) 7 835 8403

INICIO ACERCA DE SECCIONES CURSOS ELECTIVOS CONCURSOS **RESERVACIONES** @crguillarte

Reservación de locales

Inicio / Reservación de locales / Nuevo

Reservación de locales

[Guardar](#) [Cancelar](#)

Local: Tipo actividad:

Actividad: Cant. personas:

Fecha/hora actividad: Tiempo estimado (horas):

Requisitos técnicos:

| | | |
|------------------|---|--|
| Computadora | ⇌ | |
| Equipos de audio | | |

Buscar... Buscar...

[Guardar](#) [Cancelar](#)

 **Secciones**

- > Cátedras Honoríficas
- > Departamento de Bienestar Universitario
- > Manifestaciones Artísticas
- > Movimiento de Artistas Aficionados
- > Noticias
- > Proyectos Extensionistas

Contáctenos

- > Inicio
- > Acerca de
- > Cursos electivos
- > Ofertas
- > Locales
- > Administración

Dirección de Extensión Universitaria

Carretera San Antonio Km 2 1/2, Rpto Torrens (+53) 7 835 8403 extension@uci.cu

Copyright © 2019 SigLAM, Universidad de las Ciencias Informáticas.

Figura 15: Interfaz principal. Reservar local.

3.5. Conclusiones del capítulo

En este capítulo se abordaron los elementos del Sistema para la gestión de procesos en la Dirección de Extensión Universitaria, así como las pruebas realizadas al mismo y los resultados obtenidos; lo que permite concluir:

- El empleo de un estándar de codificación permitió garantizar la alta calidad, minimización errores y mayor limpieza en el código fuente. Lo que permite ser mantenido fácilmente y reutilizado por otros desarrolladores que lo necesiten.
- La implementación de la propuesta de solución facilitó la obtención de una aplicación funcional lista para su uso.
- La validación de la propuesta de solución, mediante una estrategia de pruebas de software, resultó que la misma cumple con los requisitos definidos por el cliente.

CONCLUSIONES GENERALES

El presente Trabajo de Diploma concluye con el desarrollo de un Sistema para la Gestión de Procesos en la Dirección de Extensión Universitaria, el cual sirve de apoyo a los procesos que se desarrollan en la misma.

Se destaca, además:

- El análisis y la fundamentación teórica de los principales conceptos asociados a la investigación, permitió comprender el alcance de la misma.
- El análisis de los sistemas informáticas existentes para la Gestión de Procesos en la Dirección de Extensión Universitaria, permitió definir el ambiente de desarrollo para la implementación de la propuesta de solución.
- La combinación de conocimientos adquiridos de las distintas áreas del conocimiento como son la programación, ingeniería y gestión de software, bases de datos, entre otras permitieron el análisis, diseño e implementación de la propuesta de solución.
- La definición de una estrategia de prueba, permitió comprobar el correcto funcionamiento del Sistema para la Gestión de procesos en la Dirección de Extensión Universitaria, a partir del cumplimiento de los requisitos definidos por el cliente.
- El diseño de la Base de Datos, realizado al Sistema para la Gestión de procesos en la Dirección de Extensión Universitaria, permite organizar la información generada por la DEU.
- La creación del sistema, mediante una aplicación web, contribuye a centralizar la información que se utiliza en esta área; logrando el cumplimiento del objetivo planteado.

RECOMENDACIONES

Con el cumplimiento del marcado objetivo en la presente investigación y luego de alcanzado el resultado esperado, se recomienda:

- Desplegar la propuesta de solución en la Universidad de las Ciencias Informáticas.

REFERENCIAS

- Academic.* (2012). Obtenido de Enciclopedia Universal: http://enciclopedia_universal.esacademic.com/19671/Modelado_del_Software
- Acunetix.* (5 de 12 de 2018). Obtenido de Acunetix: <https://www.acunetix.com/>
- Alonso, Y. H.; Fortún, L. L. (2014). Indicadores para evaluar la usabilidad en aplicaciones web. *1ra Conferencia Científica Internacional UCIENCIA 2014* (pág. 10). La Habana, Cuba: Universidad de las Ciencias Informáticas.
- Ambler, S. (10 de Noviembre de 2014). *Ambyssoft*. Recuperado el 20 de enero de 2017, de Ambyssoft: <http://www.ambyssoft.com/unifiedprocess/agileUP.html>
- Apache JMeter.* (5 de 12 de 2018). Obtenido de Apache JMeter: <https://jmeter.apache.org/>
- Apache Software Foundation. (20 de mayo de 2011). *The Apache HTTP Server Project*. Recuperado el 6 de diciembre de 2018, de <http://httpd.apache.org/>
- Astudillo, M., Visconti, & Hernán. (2011). *Fundamentos de Ingeniería de Software*. Universidad Técnica Federico Santa María. Recuperado el 28 de febrero de 2017
- Ávila Barrientos, E. (2014). *Formación de usuarios de la información mediante aplicaciones Web*. doi:DOI 10.5195/biblios.2014
- BELL, D. (2004). *UML basics*. Recuperado el 25 de marzo de 2017, de The component diagram: <https://www.ibm.com/developerworks/rational/library/dec04/bell/>
- CAN, C. A. (2012). *MINEDUCACIÓN*. Obtenido de SPADIES: Sistema para la Prevención de la Deserción en la Educación Superior: <https://www.mineduacion.gov.co/sistemasdeinformacion/1735/w3-article-254651.html>
- Chesneau, B. (2018). *Gunicorn*. Obtenido de <https://gunicorn.org/>
- Cuaran, J. A., & Alonso-Guerrero, J. (Noviembre de 2009). DISEÑO DE PROYECTOS CON SIG. *SIG Aplicado a la Consulta Catastral de los Predios Urbanos del Municipio de Sevilla*. Sevilla, España: UNIVERSIDAD DEL VALLE.
- Definición.* (4 de diciembre de 2018). Obtenido de <https://definicion.mx/proceso/>
- Delgado, J. (28 de junio de 2011). *Teoría de Programación*. Recuperado el 2 de diciembre de 2016, de <http://teoria-de-programacion.globered.com/categoria.asp?idcat=39>
- Django. (4 de Abril de 2017). *The Django Book*. Recuperado el 5 de diciembre de 2018, de <http://www.djangobook.com/>
- e-ABC. (2017). *e-ABC*. Recuperado el 20 de mayo de 2017, de Definición de e-Learning: <http://www.e-abclearning.com/definicione-learning>

- e-ABC. (2017). *e-ABC*. Recuperado el 20 de mayo de 2017, de Definición de e-Learning: <http://www.e-abclearning.com/queesunaplataformadeelearning>
- El blog del hogar*. (28 de septiembre de 2016). Obtenido de <http://elblogdelhogar.es/9-principios-basicos-para-una-gestion-documental-efectiva/>
- EMS Database Management Solutions, Inc. (2017). *SQL Manager*. Obtenido de <http://sqlmanager.net/en/products>
- Euphoria IT, L. (2008). *euphoriait*. Recuperado el 5 de diciembre de 2018, de <http://euphoriait.com>
- Gamma, E., Helm, R. J., & R. Vlissides, J. (2009). *Patrones de diseño*. Recuperado el 28 de febrero de 2017, de Patrones de diseño: <http://www.vico.org/pages/PatronsDisseny.html>.
- García, J. (mayo de 2005). *Lenguaje de Modelado*. Recuperado el 3 de diciembre de 2018, de <http://www.ingenierosoftware.com/analisisydiseno/uml.php>
- Guillot Jiménez, L. J. (18 de julio de 2014). SIGUA: SISTEMA INFORMÁTICO DE GESTIÓN UNIVERSITARIA. *ResearchGate*. doi:10.13140/RG.2.2.15285.47842
- Gutiérrez, E. (octubre de 2009). *JavaScript: Conceptos básicos y avanzados*. Barcelona, España: Informática Técnica. Recuperado el 16 de diciembre de 2018
- Hernandez Castillo, L. V. (abril de 2008). *Universidad Nacional Autónoma de México*. Recuperado el 4 de diciembre de 2016, de Facultad de Contaduría y Administración: http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/4/informatica_4.pdf
- Infante-Montero, S. (1 de abril de 2012). *Maestros del web*. (E. Tobar, Ed.) Recuperado el 27 de febrero de 2017, de Curso de Django para perfeccionistas: <http://www.maestrosdelweb.com/guias/#guias-django>
- ITI. (2017). *Instituto Tecnológico de Informática*. Recuperado el 29 de abril de 2017, de Testeo de estrés y carga.: <http://www.iti.es/servicios/servicio/resource/7240/index.html>.
- JetBrains Pycharm. (2017). *JetBrains*. Recuperado el 5 de diciembre de 2018, de Meet PyCharm: <http://www.jetbrains.com/pycharm/quickstart/>.
- Kasampalis, S. (2015). *Mastering Python design patterns*. Birmingham, West Midlands, United Kingdom: Packt Publishing Ltd.
- Larman, C. (2004). *UML y Patrones: una introducción al análisis*. Recuperado el 29 de enero de 2017, de UML y Patrones: <http://www.fmonje.com/UTN/ADES%20-%20208/UML%20y%20Patrones%20%202da%20Edicion.pdf>
- Marqués-Andrés, M. M. (12 de febrero de 2001). *Herramientas CASE*. Recuperado el 4 de diciembre de 2016, de Universidad Jaume I: <http://web.archive.org/web/20100928122456/http://www3.uji.es/%7Emmarques/f47/apun/node75.html>

- Martínez, J. C., Guzmán, L. A., Alarcón, V. M., & Gómez, C. M. (2013). Diagramas de navegación en aplicaciones Web. *Vol. 10(No. 2)*.
- McDonough, C. (22 de 12 de 2018). *Supervisord*. Obtenido de <http://supervisord.org/introduction.html>
- Méndez, A. V. (2010). *Metodologías de desarrollo de Software*.
- MES. (2018). *Ministerio de educación superior*. Recuperado el 12 de enero de 2019, de Ministerio de educación superior: <https://www.mes.gob.cu/es/ministerio>
- Microsoft. (2017). *Revisiones de código y estándares de codificación*. Recuperado el 10 de Abril de 2017, de <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>
- NGINX Inc. (2017). *NGINX*. Obtenido de <https://www.nginx.com/resources/wiki/>
- Nielsen, J. (2010). *Usability Engineering*. San Francisco: Editorial Morgan Kaufman.
- Ochoa P, L. A. (18 de enero de 2008). Proyecto del Máster Oficial de Software Libre. *Desarrollo de una interfaz gráfica para la creación de sentencias SQL: Caso base de datos PostgreSQL*. Barcelona, Catalunya, España: (Universitat Oberta de Catalunya. Recuperado el 20 de febrero de 2017
- Ortíz, A. (23 de 4 de 2015). Lenguaje de base de datos. *Lenguaje de manipulación de datos*.
- Pérez Porto, J., & Merino, M. (2012). Recuperado el 12 de enero de 2019, de <https://definicion.de/gestion/>
- pgAdmin. (29 de septiembre de 2016). *pgAdmin, PostgreSQL Tools*. Recuperado el 3 de diciembre de 2016, de <https://www.pgadmin.org/>
- PostgreSQL Global Development Group. (2014). *PostgreSQL*. Recuperado el 5 de diciembre de 2018, de Sistema de gestión de base de datos relacional orientada a objetos y libre: <http://www.postgresql.org/>
- Pressman, R. (2010). *Ingeniería de Software, un enfoque práctico*. Nueva York.
- Programa Nacional de Extensión Universitaria. (Noviembre de 2018). Cuba.
- Python. (2017). *Guía de estilo para el código Python – PEP 8 en Español*. Recuperado el 12 de diciembre de 2018, de www.recursospython.com: www.recursospython.com/pep8es.pdf
- Quiñones Laffita, A., Hernández Torres, I., & Cerdón González, F. (2018). Uso de la tecnología en la gestión extensionista, efectividad de la página web extensión universitaria. *III Conferencia Científica Internacional UCIENCIA 2018*.
- Ramos, M. (2014). *McGraw-Hill Interamericana de España*. Recuperado el 5 de diciembre de 2018, de <http://www.mcgraw-hill.es/bcv/guide/capitulo/8448148797.pdf>
- Rodríguez, S. (2012). *Módulo Informático para la Gestión de la Información Académica de los Estudiantes*. Matanzas.

- Rossum, G. v. (febrero de 2017). *El tutorial de Python*. Obtenido de <http://docs.python.org.ar/tutorial/pdfs/TutorialPython3.pdf>
- Saldaña, G. (2017). *Nethazard*. Recuperado el 18 de enero de 2019, de <http://blog.nethazard.net>.
- Sánchez, T. R. (6 de Marzo de 2015). Metodología de desarrollo para la Actividad productiva de la UCI v1.2. La Habana, Cuba.
- Sommerville, I. (2011). *Ingeniería de Software*. México: Pearson Educación de México.
- The Apache software foundation*. (2018). Obtenido de Apache JMeter: <http://jmeter.apache.org/>
- Torrealba, D. (8 de enero de 2016). *Principales librerías*. Recuperado el 13 de enero de 2019, de Filosofía de uso e importancia de las librerías en C++: <https://sites.google.com/site/magisterdolorestorrealba/unidad-i-introduccion-a-el-lenguaje-c/principales-libreras>
- Vigo. (28 de noviembre de 2014). *Vigo: Unión Webmaster Libres*. Recuperado el 5 de diciembre de 2018, de IDE (Entornos de Desarrollo Integrado): <https://sites.google.com/site/vigomiciudad/otras-cosa-interesantes/ide-entornos-de-desarrollo-integrado>
- Visual Paradigm. (15 de diciembre de 2016). *Visual Paradigm*. Recuperado el 14 de diciembre de 2018, de <https://www.visual-paradigm.com/support/faq.jsp>
- Zaragosa, A. (3 de Abril de 2013). Recuperado el 2 de diciembre de 2016, de Scribd: <https://es.scribd.com/document/49435158/Migrar-base-de-datos>

Anexo 1. Entrevista realizada al cliente.

Objetivo: Conocer, analizar los procesos que se realizan en la Dirección de Extensión Universitaria y determinar los requisitos que debe cumplir el sistema.

1. ¿Cuáles son los procesos que se en la Dirección de Extensión Universitaria?

2. ¿Cómo se realizan dichos procesos?

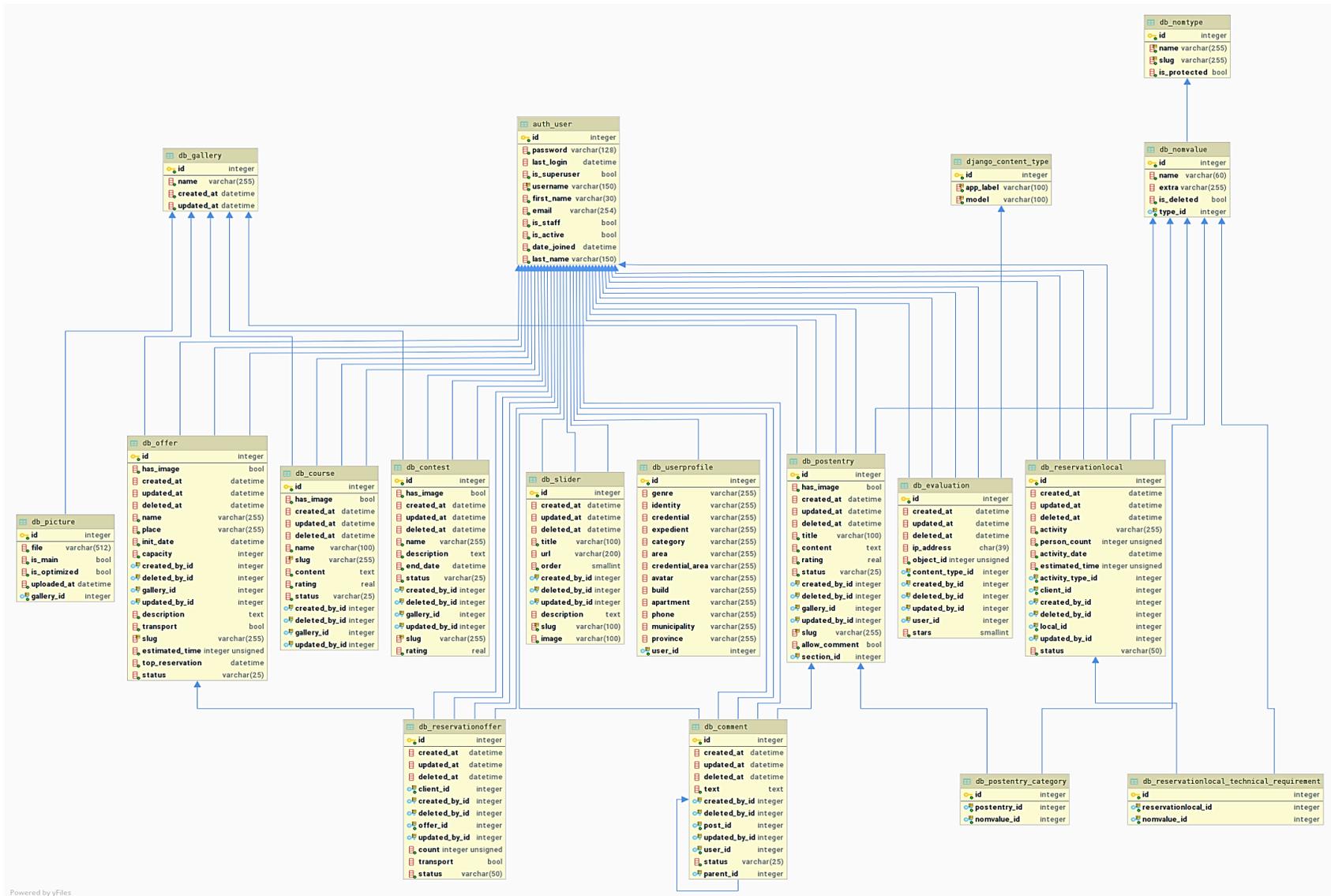
___ manual ___ automáticamente

3. ¿Es complejo realizar dichos procesos?

___ si ___ no

4. ¿Qué funcionalidades principales debería tener el sistema?

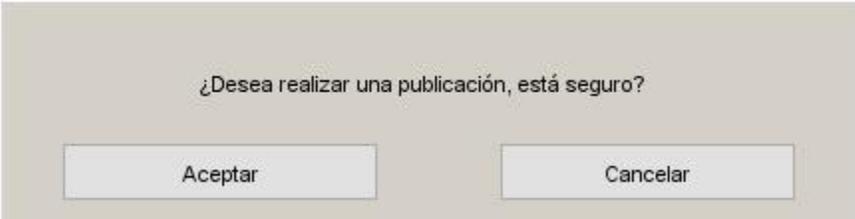
Anexo 2. Modelo de datos.

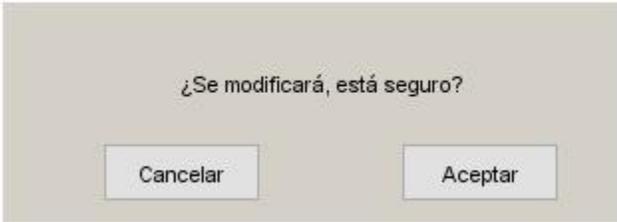


Powered by yFiles

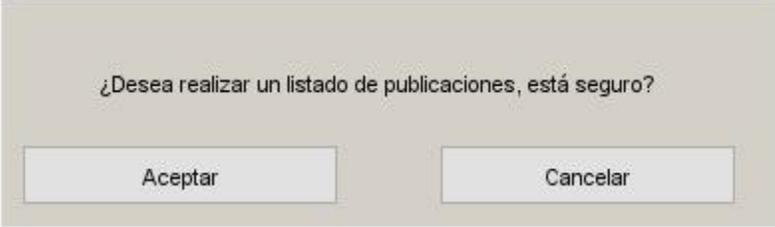
Anexo 3. Especificación de Requisitos

(Solo se muestran los requisitos con prioridad alta, los demás se pueden consultar en el Expediente de proyecto del Sistema de Extensión Universitaria)

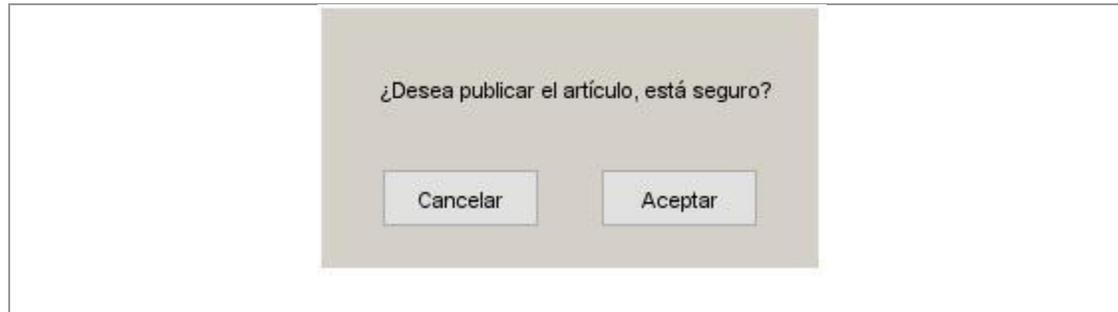
| Historia de Usuario | |
|---|----------------------------------|
| Número: HU_7 | Nombre: Crear publicación |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir crear una publicación. El sistema debe mostrar un mensaje de confirmación. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz:  | |
| Historia de Usuario | |

| | | |
|--|--------------------------------------|------------------------------|
| Número: HU_8 | Nombre: Modificar publicación | |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | | Iteración asignada: 1 |
| Prioridad: Alta | | |
| Descripción: El sistema debe permitir modificarla publicación antes creada. El sistema debe mostrar un mensaje de confirmación. | | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | | |
| Prototipo de interfaz: | | |
|  | | |

| Historia de Usuario | | |
|--|-----------------------------------|------------------------------|
| Número: HU_10 | Nombre: Listar publicación | |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | | Iteración asignada: 1 |
| Prioridad: Alta | | |
| Descripción: El sistema debe permitir mostrar un listado con las publicaciones realizadas. El | | |

| |
|---|
| sistema debe mostrar un mensaje de confirmación. |
| Observaciones: El usuario autenticado debe tener permisos de administración. |
| Prototipo de interfaz:  |

| Historia de Usuario | |
|--|---------------------------------|
| Número: HU_11 | Nombre: Publicar entrada |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir publicar el artículo. El sistema debe mostrar un mensaje de confirmación. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz: | |



Historia de Usuario

Número: HU_18 **Nombre:** Crear concurso

Programador responsable:
Claudia Rafaela Guilarte Domínguez

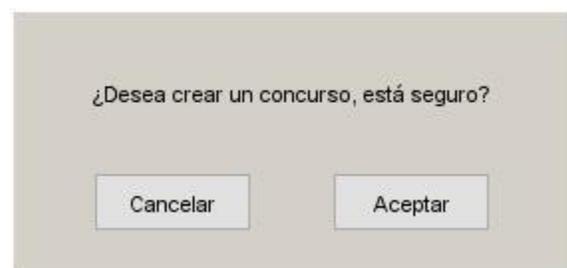
Iteración asignada: 1

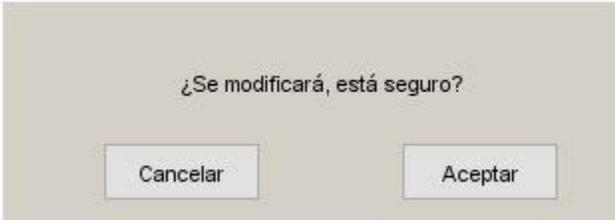
Prioridad: Alta

Descripción: El sistema debe permitir crear un concurso. El sistema debe mostrar un mensaje de confirmación.

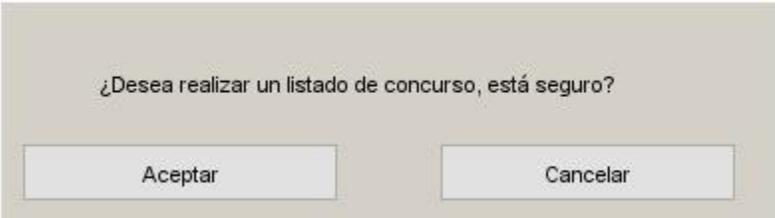
Observaciones: El usuario autenticado debe tener permisos de administración.

Prototipo de interfaz:



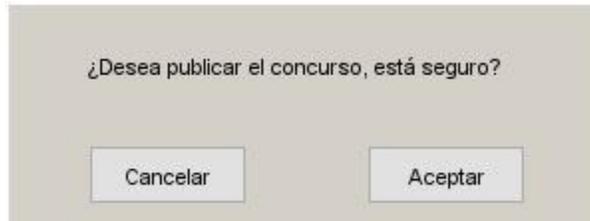
| Historia de Usuario | |
|--|-----------------------------------|
| Número: HU_19 | Nombre: Modificar concurso |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir modificar el concurso antes creado. El sistema debe mostrar un mensaje de confirmación. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz: | |
|  | |

| Historia de Usuario | |
|---|--------------------------------|
| Número: HU_21 | Nombre: Listar concurso |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |

| |
|---|
| Prioridad: Alta |
| Descripción: El sistema debe permitir mostrar un listado con los concursos publicados. El sistema debe mostrar un mensaje de confirmación. |
| Observaciones: El usuario autenticado debe tener permisos de administración. |
| Prototipo de interfaz:  |

| Historia de Usuario | |
|--|----------------------------------|
| Número: HU_22 | Nombre: Publicar concurso |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir publicar un concurso. El sistema debe mostrar un mensaje de confirmación. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |

Prototipo de interfaz:



Historia de Usuario

Número: HU_23

Nombre: Crear oferta

Programador responsable:

Claudia Rafaela Guilarte Domínguez

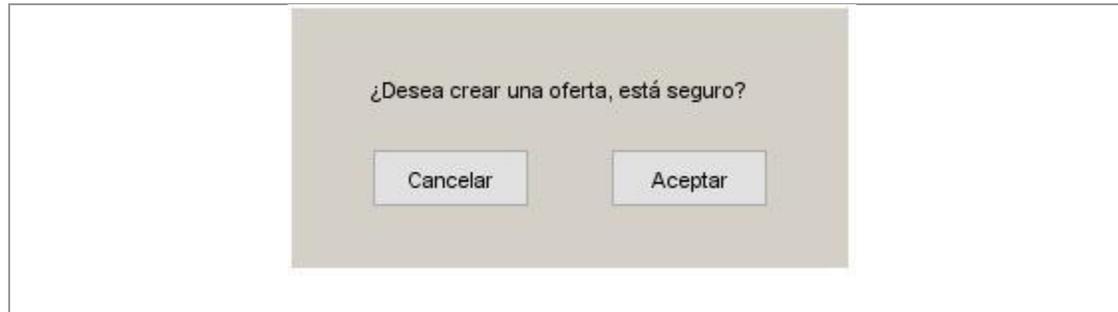
Iteración asignada: 1

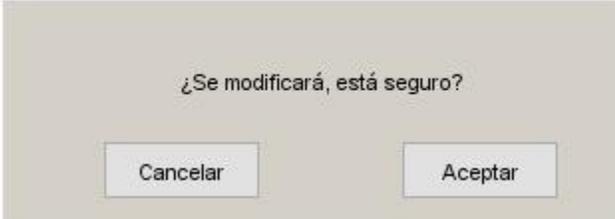
Prioridad: Alta

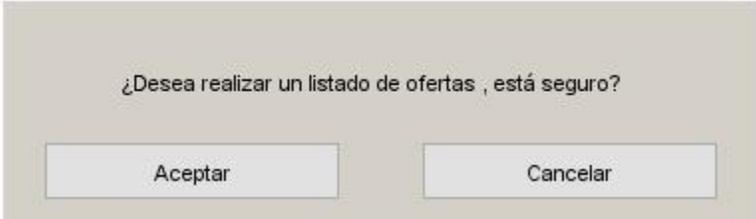
Descripción: El sistema debe permitir crear una oferta, se debe mostrar un mensaje de confirmación.

Observaciones: El usuario autenticado debe tener permisos de administración.

Prototipo de interfaz:



| Historia de Usuario | |
|--|---------------------------------|
| Número: HU_24 | Nombre: Modificar oferta |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir modificar la oferta antes creada, se debe mostrar un mensaje de confirmación. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz:  | |

| Historia de Usuario | |
|---|------------------------------|
| Número: HU_26 | Nombre: Listar oferta |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir mostrar un listado con las ofertas realizadas. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz: | |
|  <p>¿Desea realizar un listado de ofertas , está seguro?</p> <p>Aceptar Cancelar</p> | |

| Historia de Usuario | |
|---|--------------------------------|
| Número: HU_27 | Nombre: Publicar oferta |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |

| |
|--|
| Descripción: El sistema debe permitir publicar la oferta, se debe mostrar un mensaje de confirmación. |
| Observaciones: El usuario autenticado debe tener permisos de administración. |
| Prototipo de interfaz: |
|  |

| Historia de Usuario | |
|---|------------------------------|
| Número: HU_28 | Detalles de oferta |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir consultar los detalles de la oferta, se debe mostrar un mensaje de confirmación. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz: | |

| | |
|-----------|------------------|
| Nombre | Pagola la paga |
| Lugar | Teatro Karl Marx |
| Fecha | 25/4/2019 |
| Capacidad | 50 |

| Historia de Usuario | |
|---|--------------------------------|
| Número: HU_29 | Nombre: Reservar oferta |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir reservar en la oferta deseada, se debe mostrar un mensaje de confirmación. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz: | |

| | |
|---|---|
| Nombre | <input type="text" value="Pagola la paga"/> |
| Lugar | <input type="text" value="Teatro Karl Marx"/> |
| Fecha | <input type="text" value="25/4/2019"/> |
| <input type="button" value="Reservar"/> | |

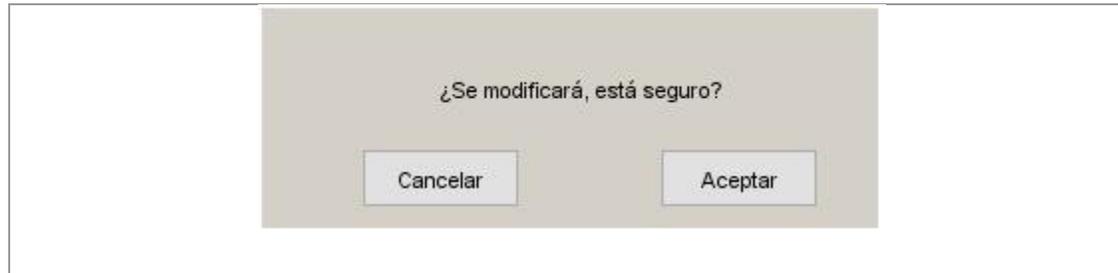
| Historia de Usuario | |
|--|-------------------------------------|
| Número: HU_30 | Nombre: Listar reservasiones |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir listar todas las reservasiones realizadas por una oferta en específico, se debe mostrar un mensaje de confirmación. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz: | |

¿Desea realizar un listado de reservaciones, está seguro?

| Historia de Usuario | |
|---|--|
| Número: HU_31 | Nombre: Crear reserva de actividades en los teatros de la universidad |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir crear una reserva de actividades en los teatros de la universidad, se debe mostrar un mensaje de confirmación. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz: | |



| Historia de Usuario | |
|---|--|
| Número: HU_32 | Nombre: Modificar reserva de actividades en los teatros de la universidad |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir modificar una reserva de actividades en los teatros de la universidad ya antes creada, se debe mostrar un mensaje de confirmación. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz: | |



| Historia de Usuario | |
|---|---|
| Número: HU_34 | Nombre: Listar reserva de actividades en los teatros de la universidad |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir mostrar un listado con las reservas de actividades en los teatros de la universidad, se debe mostrar un mensaje de confirmación. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz: | |

¿Desea realizar un listado de reserva de actividades en los teatros de la universidad , está seguro?

Aceptar

Cancelar

Historia de Usuario

Número: HU_35

Nombre: Detalles de reserva de actividades en los teatros de la universidad

Programador responsable:

Claudia Rafaela Guilarte Domínguez

Iteración asignada: 1

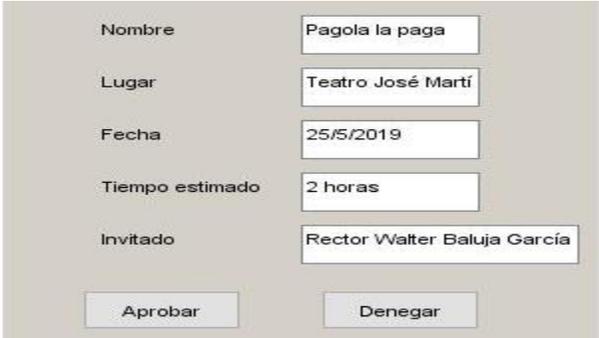
Prioridad: Alta

Descripción: El sistema debe permitir mostrarlos detalles de una reserva de actividades en los teatros de la universidad, se debe mostrar un mensaje de confirmación.

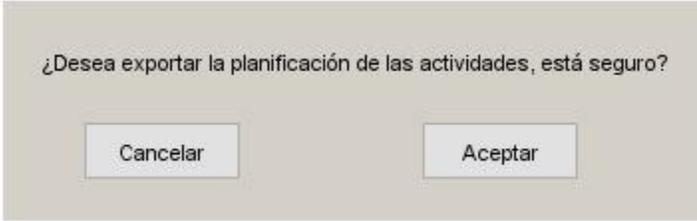
Observaciones: El usuario autenticado debe tener permisos de administración.

Prototipo de interfaz:

| | |
|-----------------|-----------------------------|
| Nombre | Pagola la paga |
| Lugar | Teatro José Martí |
| Fecha | 25/5/2019 |
| Tiempo estimado | 2 horas |
| Invitado | Rector Walter Baluja García |

| Historia de Usuario | |
|---|--|
| Número: HU_36 | Nombre: Aprobar reserva de actividades en los teatros de la universidad |
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir aprobar una reserva de actividades en los teatros de la universidad, teniendo en cuenta la disponibilidad del local a solicitar. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz: | |
|  | |

| Historia de Usuario | |
|----------------------|--|
| Número: HU_37 | Nombre: Exportar planificación por fecha, mes |

| | |
|---|------------------------------|
| Programador responsable: Claudia Rafaela Guilarte Domínguez | Iteración asignada: 1 |
| Prioridad: Alta | |
| Descripción: El sistema debe permitir exportar la planificación por fecha y mes de las reservas de actividades en los teatros de la universidad. | |
| Observaciones: El usuario autenticado debe tener permisos de administración. | |
| Prototipo de interfaz:  | |

Anexo 4: Interfaces principales

The screenshot shows a web application interface for managing publications. On the left is a dark sidebar with a menu containing: Panel de control, Seguridad, Administración, Publicaciones (highlighted), Concursos, Ofertas, Comentarios, Reservas, Galerías, and Configuraciones. The main content area is titled 'PUBLICACIONES' and includes a search bar with the text 'BUSCAR' and 'Buscar...'. Below the search bar is a table with columns for 'TITULO', 'ESTADO', and 'ACCIONES'. The table lists five publications, all with the status 'PENDIENTE'. Each row has a set of action icons: a purple eye icon, a purple image icon, a green checkmark icon, a yellow circle icon, a yellow edit icon, and a red trash icon. In the top right corner of the main area, there are two buttons: 'Mostrar eliminados' (red) and '+ Nuevo' (purple). The user's name 'Claudia Rafaela Guilarte Domínguez' and a profile picture are visible in the top right corner of the page.

| TITULO | ESTADO | ACCIONES |
|---|-----------|---|
| A conquistar el Futuro | PENDIENTE | [Icons: Eye, Image, Checkmark, Circle, Edit, Trash] |
| Acciones de motivación y orientación vocacional | PENDIENTE | [Icons: Eye, Image, Checkmark, Circle, Edit, Trash] |
| Alarcos-Teatro | PENDIENTE | [Icons: Eye, Image, Checkmark, Circle, Edit, Trash] |
| ALE.com | PENDIENTE | [Icons: Eye, Image, Checkmark, Circle, Edit, Trash] |
| Aventuras de Paquito | PENDIENTE | [Icons: Eye, Image, Checkmark, Circle, Edit, Trash] |

Figura 16: Interfaz Gestionar publicación

pages

Panel de control

Seguridad

Administración

Publicaciones

Concursos

Ofertas

Comentarios

Reservaciones

Galerías

Configuraciones

Claudia Rafaela Guillarte Domínguez

CONCURSOS

Mostrar eliminados + Nuevo

BUSCAR
Buscar...

| NOMBRE | FECHA Y HORA | ESTADO | ACCIONES |
|--------------------------|--------------------------------|-----------|---|
| Concurso del mes de Mayo | 30 de Mayo de 2019 a las 12:00 | PENDIENTE |       |

Mostrando de 1 a 1 de 1 elementos en total

Copyright © 2019 UCL. All rights reserved. [Términos y condiciones](#) SigLAM

Figura 17: Interfaz Gestionar concursos

pages

Panel de control

Seguridad

Administración

Publicaciones

Concursos

Ofertas

Comentarios

Reservaciones

Locales

Ofertas

Galerías

Claudia Rafaela Guilarte Domínguez

FORMULARIO CREAR

RESERVACIÓN DE LOCALES

NÚMERO:

SOLICITANTE: Seleccione...

TIPO ACTIVIDAD: Seleccione...

ACTIVIDAD:

CANT. PERSONAS:

FECHA/HORA ACTIVIDAD:

TIEMPO ESTIMADO:

LOCAL: Seleccione...

REQUISITOS TÉCNICOS: Seleccione...

Cancelar Guardar

Figura 18: Interfaz Gestionar reserva de actividades internas

pages

Panel de control

Seguridad

Administración

Publicaciones

Concursos

Ofertas

Comentarios

Reservaciones

Locales

Ofertas

Galerías

Claudia Rafaela Guilarte Domínguez

RESERVACIONES DE OFERTAS EXTERNAS

Mostrar eliminados + Nuevo

BUSCAR
Buscar...

| NO ^ | OFERTA | SOLICITANTE | SOLICITUD | ESTADO | ACCIONES |
|------|---|-----------------|---------------------|-----------|---|
| 1 | Show humorístico "La historia no contada de Bailando en Cuba 2" | Claudia Rafaela | 3 días, 3 horas ago | CANCELADO |     |

Mostrando de 1 a 1 de 1 elementos en total

Copyright © 2019 UCI. All rights reserved. Términos y condiciones SigLAM,

Figura 19: Interfaz Gestionar oferta