



Universidad de las Ciencias
Informáticas

Universidad de las Ciencias Informáticas
Facultad # CITEC

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

Módulo de Ayuda en línea para el Sistema de Gestión Académica
XAUCE AKADEMOS para el MINED.

Autor:

Luis David Senra Mirabal

Tutor:

Ing. Sandy Guerra Fernández



“...y si alguna vez nuestro trabajo nos pareciera bueno, debemos de luchar porque sea mejor, debemos de luchar porque sea perfecto, sabiendo de antemano que ninguna obra humana será lo suficientemente buena, ni lo suficientemente perfecta...”

Fidel Castro Ruz

Declaro ser el único autor del trabajo de diploma titulado “Módulo de Ayuda en línea para el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED”, concedo a la Universidad de las Ciencias Informáticas la autorización a hacer uso del mismo en su beneficio.

Para que conste firmo el presente documento a los ___ días del mes de noviembre del año 2022.

Luis David Senra Mirabal

Firma del autor

Datos de contacto

Tutor : Ing Sandy Guerra Fernández

Dedicatoria

A mi madre, por su amor incomparable y su confianza inquebrantable.

A mi abuelo, por su cariño infinito.

A mi hermana que a pesar de no verla a menudo siempre está ahí para mí.

A mi hermano, por ser mi ídolo y mi mayor orgullo.

A mis tías, primos y primas, por compartir mis sueños y brindarme su apoyo para hacerlos realidad.

Agradecimientos

Si algo tuve claro desde que comencé con este proceso de tesis, era que lo más difícil sería leer mis agradecimientos. A veces, caminando rumbo al docente, incluso antes de dormir se me ocurrían cosas que hoy no puedo dejar de decir. Comenzaré por aquellos que hoy no están, pero definitivamente marcaron mi persona y mi alma: mi abuela Martha la persona más dulce que he conocido, quien estuvo conmigo desde el día 1 en la primaria. A mi padre Jorge Luis gracias a ti me he convertido en el gran hombre que soy hoy. A mi familia que hoy me acompaña a pesar de no haber podido asistir. A mi abuelo Ramón te agradezco por darme ejemplo de sacrificio y perseverancia. A mi hermano por ser mi amigo mi consejero y por impulsarme al estudio desde pequeño. A mi mamá por ser la persona más importante en mi vida, por entenderme y apoyarme siempre en todas mis decisiones sin ti no sabría que fuera de mí hoy día. A esas amistades que se volvieron familia de aquel Fi35 el mejor grupo de nuestra FICl. A todos los compañeros de apartamento del 96103. A Alfredo y Osmel que sé que en un año se estarán graduando. A Danay que se convirtió en una hermana dentro de la escuela. A mi tutor Sandy por todo el apoyo, la confianza y los consejos brindados. Al oponente Luis Miguel por su labor y al tribunal por sus recomendaciones para lograr que el trabajo de diploma tenga la calidad con la que hoy se presenta ¡A todos, muchas gracias!!!

Resumen

El Ministerio de Educación de Cuba busca automatizar e informatizar todos los procesos académicos de las distintas escuelas pertenecientes al sistema educacional. Para esto plantea la integración del Sistema de Gestión Académica XAUCE AKADEMOS. A pesar de la fortaleza de sus funcionalidades este no cuenta con una guía de apoyo para los usuarios que interactúan con él.

El presente trabajo que lleva por título “Módulo de Ayuda en línea para el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED” pretende brindar la información referente al funcionamiento del módulo que conforma la ayuda en el sistema para facilitar y mejorar el trabajo de los usuarios con el mismo. Este documento recoge el resultado de todo el trabajo realizado para la elaboración de la ayuda en línea, incluyendo el estudio del estado del arte de aplicaciones con similares objetivos existentes en el mundo, el estudio y definición de las características del módulo, así como la planificación, el diseño, la implementación y prueba del producto.

Palabras clave: Ayuda en línea, módulo, preguntas frecuentes, Sistema de Gestión.

Abstract

The Cuban Ministry of Education seeks to automate and computerize all the academic processes of the different schools belonging to the educational system. For this purpose, it proposes the integration of the Academic Management System XAUCE AKADEMOS. Despite the strength of its functionalities, it does not have a support guide for the users who interact with it.

The present work entitled "Online Help Module for the Academic Management System XAUCE AKADEMOS for MINED" aims to provide information on the operation of the module that makes up the help in the system to facilitate and improve the work of users with it. This document gathers the result of all the work carried out for the development of the online help, including the study of the state of the art of applications with similar objectives existing in the world, the study and definition of the characteristics of the module, as well as the planning, design, implementation and testing of the product.

Keywords: Frequently asked questions(FAQ's), Management System, Online help, Module.

Índice

| | |
|---|-----------|
| Introducción..... | 11 |
| CAPÍTULO I: Fundamentos Teóricos..... | 16 |
| 1.1 Conceptos asociados al dominio del problema..... | 16 |
| 1.1.1 Características de los Sistemas de Ayuda en Línea..... | 17 |
| 1.1.2 Procedimiento para el desarrollo de la Ayuda en Línea..... | 17 |
| 1.2 Situación actual a nivel mundial..... | 19 |
| 1.2.1 Sistemas de Ayuda en Línea en Cuba..... | 20 |
| 1.3 Metodologías de desarrollo de software..... | 21 |
| 1.3.1 Metodología de Proceso Unificado Ágil (AUP)..... | 22 |
| 1.3.2 Metodología AUP-UCI..... | 23 |
| 1.4 Herramientas y tecnologías asociadas al desarrollo del componente..... | 26 |
| 1.4.1 Lenguajes..... | 26 |
| 1.4.2 Herramientas..... | 27 |
| 1.4.3 Marco de Trabajo..... | 28 |
| 1.5 Conclusiones parciales..... | 29 |
| CAPÍTULO II: Propuesta de Solución..... | 30 |
| 2.1 Propuesta del Módulo de Ayuda en Línea..... | 30 |
| 2.2 Modelado del negocio..... | 31 |
| 2.3 Requisitos del Software..... | 32 |
| 2.3.1 Definición de las técnicas de obtención de requisitos..... | 32 |
| 2.3.2 Definición de Requisitos Funcionales..... | 32 |
| 2.3.3 Definición de Requisitos No Funcionales..... | 34 |
| 2.3.4 Descripción de Requisitos Funcionales..... | 36 |
| 2.4 Arquitectura y Diseño de Software..... | 41 |
| 2.4.1 Patrones de diseño..... | 42 |
| 2.5 Conclusiones Parciales..... | 44 |
| CAPÍTULO III: Validación de la Propuesta de Solución..... | 45 |
| 3.1 Estándares de codificación..... | 45 |
| 3.2 Validación de requisitos..... | 47 |
| 3.2.1 Técnicas de validación de requisitos..... | 48 |
| 3.3 Pruebas de software..... | 48 |
| 3.3.1 Niveles de prueba..... | 49 |
| 3.3.2 Resultados obtenidos..... | 50 |

Conclusiones..... 55
Recomendaciones..... 56
Referencias bibliográficas..... 57
Anexos..... 62

Índice de ilustraciones

Ilustración 1 Diagrama de Paquetes correspondiente al Modelo-Vista-Plantilla.....43
Ilustración 2 Patrón GRASP Experto.....44
Ilustración 3 Patrón GRASP Controlador.....44
Ilustración 4 Ejemplo del estándar CamelCase.....46
Ilustración 5 Ejemplo de CSS.....47
Ilustración 6 Ejemplo de HTML.....47
Ilustración 7 Ejemplo de Django.....48
Ilustración 8 Prueba unitaria de la clase TestUrls.....51
Ilustración 9 Resultado mostrado en consola.....52
Ilustración 10 Gráfica representativa de las no conformidades.....53

Índice de tablas

Tabla 1 Personas que interactúan con la ayuda en línea.....30
Tabla 2 Listado de requisitos funcionales.....33
Tabla 3 Listado de requisitos no funcionales.....34
Tabla 4 Descripción de Requisitos Funcionales.....36
Tabla 5 Descripción detallada del RF1.1 Adicionar Pregunta.....40
Tabla 6 No conformidades detectadas mediante las pruebas de sistema.....52
Tabla 7 Diseño de caso de prueba Adicionar pregunta.....54

Introducción

El Ministerio de Educación (MINED) es un sistema integrado por instituciones de educación, que rige metodológicamente toda la estructura escolar cubana. Este sistema, bajo la dirección del Partido Comunista de Cuba (PCC) y el Gobierno, tiene como misión esencial, dirigir, ejecutar y controlar la aplicación de la política del Estado y el Gobierno en la actividad educacional, excepto en la Educación Superior (1). Además de formar y consolidar 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 de todos los estudiantes y profesionales egresados de él.

Durante las últimas dos décadas el país se ha concentrado en la informatización de todos sus sectores, siendo el área de educación una de las que ha acaparado los mayores focos. Debido al avance de las tecnologías se hace necesario informatizar gran parte de los procesos que se realizan a día de hoy manualmente.

El MINED como órgano principal de la educación lidera este proceso de renovación que existe en las filas institucionales de la nación. El nuevo paso que ha dado es la integración de XAUCE AKADEMOS, una herramienta multiplataformas que contribuye al perfeccionamiento de los procesos académicos de una institución. Su uso permite el desarrollo coherente de una estrategia organizacional en los diferentes niveles de la estructura educacional con la que cuenta el Ministerio de Educación de la República de Cuba.

XAUCE AKADEMOS ha sido por varios años el sistema que ha dado respuesta a los cambios y necesidades del personal de la UCI que interactúa diariamente con él. Actualmente sufre una serie de cambios estructurales que van desde la base hasta los niveles superiores.

Dada a la ardua tarea que suponía desarrollar un software de este tipo, fue necesario emplear las mejores prácticas de desarrollo disponibles, desde el punto de vista arquitectónico. Entre las opciones más acordes, para dar solución a las necesidades del cliente, teniendo en cuenta las características técnicas de la infraestructura del MINED, se encontraba la adopción de una arquitectura orientada a microservicios, la cual permite un desarrollo desacoplado de componentes con independencia del lenguaje o tecnología, y que pueden ser integrados a la solución con gran facilidad, luego de rebasar satisfactoriamente un proceso de pruebas. De manera general todas estas transformaciones y funcionalidades de la nueva versión de AKADEMOS, están dirigidas a la creación de un nuevo producto que sea adaptable a cualquier centro de estudios y, a la vez, se ajuste a las políticas de Software Libre llevadas a cabo por nuestro país.

Introducción

La nueva estructuración del software es directamente proporcional a un aumento considerable del nivel de complejidad del mismo añadiendo que para, los nuevos usuarios y administradores, a los que va dirigido, es su primera experiencia ante una plataforma como esta, por lo que uno de los campos menos explotados en estas plataformas como es el diseño creación y gestión de un módulo de Ayuda en Línea se hace sumamente necesario.

La Ayuda en Línea es un tipo de sistema de apoyo que proporciona una información breve y concisa para resolver un problema puntual durante el trabajo con el software en cuestión.

“Un sistema de asistencia en línea es un software que presta ayuda al usuario en la utilización de alguna funcionalidad del programa mientras éste se encuentra directamente conectado con el ordenador. Esta ayuda se presentará en la forma de asistencia de órdenes, mensajes de error, documentación en línea o manuales de aprendizaje. La ayuda en línea estaría separada del sistema sobre el cual se ofrece o integrada en la propia arquitectura del programa”

(Frankel and Balci, 1989) (2)

La UCI como universidad cubana productora de software, para lograr un resultado óptimo en todos sus proyectos y observando la importancia que tienen, desarrolla la integración de Módulos de Ayuda en Línea como soporte tecnológico para todos los usuarios. La plataforma XAUCE AKADEMOS no se encuentra exenta de la necesidad de incorporar este tipo de herramienta por la situación descrita a continuación:

El Sistema de Gestión Académica XAUCE AKADEMOS cuenta con una interfaz gráfica, la cual es novedosa para todos los usuarios que operan el software. El mismo agrupa nuevos módulos, componentes y aplicaciones de las cuales los usuarios que lo operan no poseen, en su mayoría, los conocimientos informáticos necesarios para su utilización. Varias funcionalidades similares no se realizan de forma similar, lo cual aumenta la curva de aprendizaje y dificulta la experiencia del usuario. Gran cantidad de módulos son dependientes y heredan de otros lo que conlleva al sistema a tener un alto grado de complejidad que hace disminuir la comprensión del mismo para el personal al cual está destinado. Debido al gran cúmulo de información que este gestiona se producen demoras en diferentes etapas de procesos y no brinda respuesta de su realización, esto genera que disminuya la confianza por parte de los clientes en el uso de la plataforma. Para realizar los procesos no sólo se necesitan conocimientos del negocio sino conocimientos de las funcionalidades, flujo de información y navegabilidad del nuevo software. La mayoría de los usuarios posee edad avanzada, años de experiencia en la institución trabajando con otro sistema o manualmente, lo cual propicia la resistencia al cambio. Por tanto, se hace presente la necesidad

Introducción

de que exista una forma de retroalimentación entre cliente-equipo técnico para fomentar la comunicación en tiempo real entre ellos, lo cual le brindara al usuario una mejor experiencia y una mayor comprensión de la plataforma, permitiéndoles interactuar de forma inmediata y esclarecer sus dudas.

Debido a la situación planteada surge como **problema a resolver**: ¿Cómo garantizar el proceso de ayuda técnica e informativa en el sistema XAUCE AKADEMOS?

Para solucionar el problema planteado se tiene como **objetivo general**: Desarrollar un Módulo de Ayuda en Línea para el software XAUCE AKADEMOS para el MINED.

Como **objeto de estudio** tendremos: Módulos de Ayuda en Línea.

Nuestro **Campo de acción**: Módulos de Ayuda en Línea en Sistemas de Gestión.

Para la presente investigación se trazaron los siguientes **objetivos específicos**:

1. Realizar un estudio teórico y del arte sobre los Módulos de Ayuda en Línea.
2. Efectuar el procedimiento de realización del Módulo de Ayuda en Línea del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED.
3. Implementar y Validar el Módulo de Ayuda en Línea del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED.

Para dar cumplimiento a los objetivos propuestos se determina las siguientes **tareas**:

- Estudio del estado del arte, comparación y selección de las metodologías de desarrollo de software.
- Estudio del estado del arte de los principales Módulos de ayuda en línea en la actualidad.
- Estudio del estado del arte y selección de los sistemas de gestión de contenidos más usados en la actualidad que permitan una correcta implementación de ayudas en línea.
- Estudio del estado del arte y selección del lenguaje de programación a utilizar en la implementación de la ayuda en línea.
- Estudio del estado del arte de los principales Sistemas Gestores de Bases de Datos en la actualidad.
- Identificación de los procesos del negocio del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED.
- Realización del levantamiento de requisitos.
- Análisis de la estructura de los módulos del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED para la implementación del módulo ayuda en línea.
- Obtención de los artefactos que propone la metodología elegida, que permitan un correcto diseño de la ayuda en línea.

- Implementación del módulo de ayuda en línea para el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED.
- Validación de la propuesta de solución.
- Instalación del módulo ayuda en línea obtenida.

Posibles resultados:

Módulo de Ayuda en Línea para el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED que brinde soporte a las mayores dudas de los usuarios.

Métodos Científicos 3:

Métodos teóricos:

- Análisis – Síntesis: Consiste en el análisis de los documentos para identificar y separar los elementos fundamentales. Este método se utilizó para estudiar aquellos documentos que contienen información que se incluye en la ayuda en línea.
- Histórico: Está vinculado al conocimiento de las distintas etapas de los objetos en su sucesión cronológica, para conocer la evolución y desarrollo del objeto. Se pone de manifiesto en el estudio del sistema que se trata, cómo ha evolucionado XAUCE AKADEMOS desde su origen.

Métodos Empíricos

- Observación científica: El investigador conoce el problema y el objeto de investigación, estudiando su curso natural. Se utiliza en todo momento en las entrevistas con el cliente y en el estudio de los módulos del sistema.
- Consulta a especialistas: Se empleó para comprobar la necesidad y la funcionalidad práctica de la ayuda en línea que se propone en la presente investigación.
- Entrevistas: Propiciaron recoger las opiniones que sobre el tema de investigación poseen distintos especialistas conocedores del tema tratado.

La estructura del documento que soporta la investigación está compuesta por 3 capítulos, además de agradecimientos, dedicatoria, introducción, conclusiones, recomendaciones, referencias bibliográficas, bibliografía y anexos.

Capítulo I: Fundamentación teórica de la investigación.

Se expone la fundamentación teórica del trabajo, abordando todos los elementos teóricos que sustentan el problema científico, los objetivos del trabajo y se realiza un análisis de las tendencias de

Introducción

las tecnologías y metodologías actuales en el desarrollo de software. Se fundamentan las tecnologías y herramientas con las cuales se debe desarrollar la ayuda, así como la metodología más adecuada para guiar el desarrollo de la misma.

Capítulo II: Propuesta de solución.

Se presenta la modelación del negocio realizada, se describirán los procedimientos para la realización del Módulo Ayuda en Línea, se exponen los requisitos funcionales y no funcionales del módulo, se detallan los artefactos generados en las fases de Planificación y Diseño, teniendo especial atención en las Historias de Usuario.

Capítulo III: Validación de la propuesta de solución

Se describen las tareas relacionadas a la fase de Implementación y las Pruebas de Aceptación realizadas a solicitud del cliente en la fase de Prueba.

CAPÍTULO I: Fundamentos Teóricos

Los Sistemas de Ayuda son usados para el entendimiento de un producto, mejorar su aprendizaje y rendimiento en entornos informáticos. Dichos sistemas son de gran importancia pues guían y facilitan al usuario en su interacción con un software.

El objetivo de este capítulo es proporcionar distintos aspectos teóricos como son los principales conceptos relacionados con la investigación. Además, se encuentra el análisis de las herramientas de Ayuda en Línea a nivel mundial hasta la actualidad. Se incluye también las descripciones de las herramientas, tecnologías, metodologías y lenguajes de programación que se utilizarán en el desarrollo de la solución.

1.1 Conceptos asociados al dominio del problema

En línea:

Del Sitio AdLógica:

Término general que hace referencia a cualquier elemento conectado a una red de comunicación o transmitido a través de ella. (4)

Ayuda en línea:

Del Mini Diccionario Informático de Carlos Pes1:

A veces, las aplicaciones informáticas ofrecen una ayuda en línea, en donde se explica al usuario los pasos que debe seguir para instalar dicha aplicación, configurarla, utilizarla. (5)

Help Desk (Mesa de Ayuda) o Departamento de Soporte Técnico:

Aplicaciones o sistemas encargadas de proporcionar servicios eficientes y oportunos de tecnologías de información a una organización, a través de un sencillo espacio físico, una línea telefónica, una dirección de correo electrónico o cualquier otro canal de comunicación. (6)

En los casos anteriores se pueden encontrar puntos comunes. El más importante de ellos es que un Sistema de Ayuda en Línea: constituye una aplicación web con el objetivo de brindar atención y solución a las necesidades de los usuarios para configurar, instalar o utilizar una aplicación informática.

1.1.1 Características de los Sistemas de Ayuda en Línea

Desde el punto de vista del usuario es necesario tener en cuenta ciertas características que deben poseer los Sistemas de Ayuda en Línea para aprovechar su utilidad al máximo. (7)

- Disponibilidad: Se puede acceder a la ayuda en cualquier momento de la interacción en el sistema sin tener que salir de la aplicación.
- Precisión y detalle: La ayuda de un sistema debe ser precisa y detallada, y uno de los problemas más importantes para conseguirlo es la constante actualización de las versiones de software. Si las funcionalidades del software no se encuentren vinculadas a la aplicación, el usuario quedará desilusionado. Hay que tener en cuenta que, cuando se efectúa el diseño, no se conoce qué partes de la ayuda serán las más utilizadas, por tanto, se debe prestar la misma importancia a cada una de ellas.
- Flexibilidad: Muchos sistemas de ayuda son rígidos, pues no tienen en cuenta el nivel de experiencia de la persona que lo está utilizando, o no tiene en cuenta el contexto en que ésta trabaja. Un sistema de ayuda flexible permite que los diferentes tipos de usuarios interactúen de una manera apropiada según sus necesidades.
- Robustez: Los Sistemas de Ayuda se utilizan frecuentemente cuando el usuario se encuentra en dificultades, por tanto, es importante su robustez, especialmente en aquellos casos en que las aplicaciones no se comportan correctamente.
- Consistencia: Todas las partes de la ayuda y la documentación han de ser consistentes en términos de contenidos, terminología y estilo. La consistencia tiene un refuerzo en la presentación. No obstructiva. Es muy importante que la ayuda no se interponga en el uso normal de la aplicación, pues esto decepcionaría al usuario.

1.1.2 Procedimiento para el desarrollo de la Ayuda en Línea.

Para desarrollar la Ayuda en Línea se deben seguir una serie de pasos con secuencia lógica para garantizar que los objetivos de la misma sean cumplidos. (8)

Planificación del contenido: durante esta etapa se define qué se va a tratar en la Ayuda, qué aspectos se incluirán y qué no, además se debe escoger la información a poner de acuerdo con

las especificaciones hechas. Se debe evaluar la cantidad de tiempo que se dispone contra la de tiempo que se requiere para su preparación; ya que muchos proyectos comienzan a crecer en cuanto a sus especificaciones sin que haya un trabajo ordenado de delimitación del contenido de la ayuda, y esto lleva al fracaso. De manera general se obtiene toda la información relativa a las tareas y procedimientos del software que van a ser documentados.

Estructuración de la información: cuando ya se cuenta con la información que se desea incluir, se inicia un proceso de clasificación. Se realiza una clasificación de tipo jerárquica, dividiendo la información en módulos, submódulos y secciones. Cada módulo contiene submódulos que a su vez están compuestos por secciones, cada sección es una página de la ayuda. Se define el entorno y la audiencia de la aplicación, así como su relación con la documentación que será utilizada.

Diseño de la ayuda: el diseño de la Ayuda en Línea está centrado fundamentalmente en el usuario y las tareas que este realiza en la aplicación. Se deben tener en cuenta varios aspectos:

- Definición de guías de contenidos detalladas: expandir el contenido en módulos, submódulos y secciones, delimitando la información tanto textual como gráficos
- Determinación de tópicos: estimar el número total de las páginas que posee la Ayuda, incluyendo los nombres de los módulos, submódulos y secciones.
- Especificaciones de estilo y diseño gráfico: se deben desarrollar guías de estilo para crear consistencia entre todas las páginas. Un estilo gráfico adecuado es muy importante para hacer que el usuario que visite la página se sienta cómodo. Otro aspecto importante es la parte gráfica, se debe intentar en lo posible mantener una coherencia gráfica, por ejemplo, un fondo agradable o una distribución interesante de imágenes.
- Riesgos y dependencias: todas las actividades de un proyecto pueden sufrir cambios respecto a cómo fueron planificadas inicialmente, por lo que es importante las revisiones periódicas en busca de cambios.
- Testeo: por último, es necesario revisar que exista coherencia general en la ayuda, revisar la ortografía y redacción de las páginas, separar páginas muy extensas en más pequeñas, eliminar los vínculos rotos que no conduzcan a ninguna parte.

1.2 Situación actual a nivel mundial

Existen en el mundo disímiles sistemas de ayuda en línea que brindan información sobre diversas aplicaciones informáticas. Todos tienen como objetivo principal ofrecer la información necesaria para que los usuarios Web estén en capacidad de utilizar en forma eficiente las potencialidades que le ofrece dicha aplicación para facilitar sus labores diarias. Dentro de estos sistemas podemos mencionar algunos:

▣ Zoho Desk

Zoho Desk es una solución en la nube que pertenece a Zoho Corporation. Las características clave incluyen la gestión de tickets de soporte, un portal de atención al cliente, gestión de contratos y creación de informes. Zoho Desk recopila interacciones de varias fuentes (correo electrónico, teléfono, chat, redes sociales, portal de autoservicio, foros y formularios) y las presenta en un único lugar. Tareas como la asignación de tickets, escalamientos de servicio, reglas de notificación y acciones basadas en el tiempo pueden automatizarse mediante el establecimiento de reglas de flujo de trabajo. La solución proporciona informes personalizables y programados, y un panel gráfico para analizar la satisfacción del cliente. (9)

▣ Jira Service Desk

Otro sistema de Help Desk disponible actualmente es Jira Service Desk, su propuesta consiste en reunir bajo una misma plataforma a desarrolladores, personal operativo y de soporte para responder de inmediato a la solicitud de los clientes. Este software figura entre los más competitivos del mercado por su efectividad cuando de resultados se trata. Como software, Jira Service Desk puede integrarse con otras herramientas para una asistencia aún más completa. Su sistema puede ser autogestionado con iCloud y Windows, o almacenar la información en la Nube para un rendimiento óptimo de tu sitio empresarial. (10)

▣ Servicetonic

▣

Servicetonic es empleado por varias compañías líderes en el mercado para gestionar su atención al cliente con resultados satisfactorios. Su software Help Desk está basado en la nube, por lo que no requiere instalación en los servidores de la empresa y dispone de un equipo profesional de

apoyo que ayudará a la misma con su implementación. El servicio de *ticketing*¹ de Servicetonic es automatizado, lo que permite gestionar eficazmente cada solicitud. Paralelo a ello, la atención es multicanal (web, voz, chat, QR) para brindar mayores opciones al cliente y pone a disposición de tu negocio un completo portal de autoservicio, para quienes deseen realizar consultas a través de la web. (11)

1.2.1 Sistemas de Ayuda en Línea en Cuba

En Cuba también se han desarrollado sistemas de ayuda en línea para satisfacer las necesidades de los usuarios. Entre ellos se pueden destacar sitios como el de Infomed.

Ayuda en línea de Infomed:

Infomed es el Portal de Salud Cubano y la red de personas e instituciones que comparten el propósito de facilitar el acceso a la información de salud en el país. Para facilitar la navegación por el sitio se incluyó una ayuda al usuario, la cual permite conocer la información que podrá consultar en la Biblioteca Virtual de Infomed, así como las demás funcionalidades de navegación por la web, cómo realizar una búsqueda, dónde informarse sobre los principales eventos que se desarrollan en el sector de la salud en nuestro país, cómo se distribuye la información en el sitio entre otros temas de interés. (12)

Sistemas de Ayuda en Línea usados en la UCI

Centro de Soporte de Universidad de las Ciencias Informáticas:

Para una mejor navegación y experiencia de los clientes en el portal oficial de la UCI está implementado un sistema de ayuda en línea llamado Centro de Soporte, el cual ofrece servicios de asistencia técnica a las aplicaciones y servicios informáticos desarrollados por la Universidad de las Ciencias Informáticas, con calidad y eficiencia; a partir de una correcta gestión. (13). Entre sus funciones se encuentran:

¹ El sistema de ticketing es una herramienta que supervisa y documenta las interacciones del cliente con los agentes del servicio de atención al cliente y permite gestionar de forma eficiente el flujo de contactos entrantes, ya tanto si provienen de sitios del cliente como de redes sociales o foros en línea

- Asumir el soporte y despliegue de las aplicaciones desarrolladas por la Red de Centros a nivel nacional e internacional.
- Mantener asociaciones estratégicas con los clientes, basados en la mejora continua para incrementar la calidad de los productos y servicios.

Ayuda en Línea de Picta:

El Sistema de Ayuda en Línea integrado a Picta brinda al usuario varias posibilidades para instruirse y que este mismo tenga una mejor experiencia en su navegación por el sitio. Este sistema está constituido por:([14](#))

- Preguntas frecuentes (FAQ's): que no es más que un listado de preguntas o dudas que surgen comúnmente en un contexto determinado y sobre un tema específico.
- Términos de uso: Se agrupan los términos de mayor importancia y de mayor grado de comprensión para los usuarios.
- Correo electrónico: Un correo electrónico para la comunicación directa con el equipo técnico.

1.3 Metodologías de desarrollo de software

Una metodología es una colección estructurada de procedimientos que ayudan a los desarrolladores de software en sus proyectos, ofreciendo una guía para la toma de decisiones, así como para planificarlo, gestionarlo, controlarlo y evaluarlo. La elección de la metodología a emplear es clave durante el desarrollo de un software por sus implicaciones en lo referente a efectividad, eficiencia y desempeño del producto, costo y el tiempo de desarrollo, métodos de control de calidad y de pruebas, los cuales deben ajustarse a las particularidades de cada metodología. ([15](#))

Una metodología de desarrollo de software es similar a una receta de cocina. De la misma manera que una receta le enseña a uno cómo cocinar una comida, un método de desarrollo de software enseña cómo construir un producto de software. Las metodologías aumentan los esfuerzos para mejorar la calidad de los productos al mejorar los procesos que producen los productos. Se requiere un estándar o método formalmente definido para controlar los procesos de desarrollo. Las diferentes metodologías de desarrollo de software tienen fortalezas y debilidades. Existen las metodologías ágiles y tradicionales. La elección de qué método usar realmente depende de los objetivos que una empresa quiera alcanzar.

Metodologías tradicionales de desarrollo del software:

Estas metodologías imponen una disciplina de trabajo sobre el proceso de desarrollo del software, buscando conseguir un software más eficiente y predecible. Se focalizan principalmente en la documentación, planificación y procesos; este enfoque es considerado tradicional por ser el primero que se empleó para desarrollar software. (16)

Ejemplos de estas metodologías son:

- Proceso Racional Unificado (RUP)
- Microsoft Solution Frame (MSF).

□

Metodologías ágiles de desarrollo del software

Hoy en día, el mundo empresarial opera en un entorno global que cambia rápidamente; por ende, se debe responder a las nuevas necesidades y oportunidades del mercado, teniendo en cuenta que el software es partícipe de casi todas las operaciones empresariales. Se debe desarrollar soluciones informáticas de manera ágil para poder dar una respuesta de calidad a todo lo necesario. (17)

Las metodologías ágiles presentan como principal particularidad la flexibilidad, los proyectos en desarrollo son subdivididos en proyectos más pequeños, incluye una comunicación constante con el usuario, son altamente colaborativos y es mucho más adaptable a los cambios. De hecho, el cambio de requerimientos por parte del cliente es una característica especial, así como también las entregas, revisión y retroalimentación constante.

Entre las más notables metodologías de desarrollo ágil, se encuentran:

- Scrum (Muy popular en emprendimientos)
- Programación extrema (XP)
- Crystal Clear Mobile-D (ágil y extrema para móviles)
- Proceso Unificado Ágil (AUP)

1.3.1 Metodología de Proceso Unificado Ágil (AUP)

El Proceso Unificado Ágil de Scott Ambler en inglés Agile Unified Process (AUP) es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (test driven development - TDD), Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad. (18)

El proceso unificado (Unified Process o UP) es un marco de desarrollo software iterativo e incremental. A menudo es considerado como un proceso altamente ceremonioso porque especifica muchas actividades y artefactos involucrados en el desarrollo de un proyecto software. Dado que es un marco de procesos, puede ser adaptado y la más conocida es RUP (Rational Unified Process) de IBM.

AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos.

El proceso AUP establece un Modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP.

1.3.2 Metodología AUP-UCI

Al no existir una metodología de software universal en los proyectos desarrollados en la UCI, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable, se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva. Con dicha adaptación se logra estandarizar el proceso de desarrollo de software, dando cumplimiento además a las buenas prácticas que define CMMI-DEV ²v1.3 las cuales se centran en el desarrollo de productos y servicios de calidad. (19)

Descripción de las Fases de AUP-UCI (19)

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

² Modelo de Madurez de Capacidad Integrado para el Desarrollo

- **Ejecución:** En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
- **Cierre:** En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Disciplinas de la Metodología Variación AUP-UCI (19)

- **Modelado de negocio:** Destinada a comprender los procesos de negocio de una organización. Se define cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito. Para modelar el negocio se proponen las siguientes variantes: Casos de Uso del Negocio (CUN), Descripción de Proceso de Negocio (DPN) y Modelo Conceptual (MC). A partir de las variantes anteriores se condicionan cuatro escenarios para modelar el sistema en la disciplina Requisitos.
- **Requisitos:** El esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto. Existen tres formas de encapsular los requisitos: Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por proceso (DRP), agrupados en cuatro escenarios condicionados por el Modelado de negocio.
- **Análisis y diseño:** Los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Además, en esta disciplina se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos que el de análisis.
- **Implementación:** En la implementación, a partir de los resultados del Análisis y Diseño se construye el sistema.
- **Pruebas internas:** Se verifica el resultado de implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberada. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posible componentes de prueba ejecutables para automatizar las pruebas

- Pruebas de liberación: Diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.
- Pruebas de Aceptación: Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

Descripción de los escenarios (19)

A partir de que el Modelado de negocio propone tres variantes a utilizar en los proyectos (CUN, DPN o MC) y existen tres formas de encapsular los requisitos (CUS, HU, DRP), surgen cuatro escenarios para modelar el sistema en los proyectos, manteniendo en dos de ellos el MC, quedando de la siguiente forma:

- Escenario No 1: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta respectivamente, donde la atención se centra en cómo el usuario va a utilizar el sistema.
- Escenario No 2: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio.
- Escenario No 3: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad.
- Escenario No 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos.

La realización, desarrollo e implementación del Módulo de Ayuda en Línea para el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED se va a regir por el **escenario 3** de la

metodología AUP-UCI ya que presenta un negocio de gran complejidad. Además, es el escenario por el cual se ha construido todo el sistema.

1.4 Herramientas y tecnologías asociadas al desarrollo del componente

A continuación, se realiza una descripción de los lenguajes seleccionados para el desarrollo del Módulo de Ayuda en Línea para el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED.

1.4.1 Lenguajes

Los lenguajes de programación son una sintaxis codificada usada por los programadores para comunicarse con la computadora. Es el único lenguaje que entienden tanto los programas de software como los de hardware. El lenguaje le permite al usuario dictar los comandos que la computadora debe entender para procesar los datos, estos vienen en varios tipos que emplean conjuntos de sintaxis diferentes. (20)

HTML 5

HTML (Lenguaje de Marcas de Hipertexto, del inglés *HyperText Markup Language*) es el componente más básico de la Web. Define el significado y la estructura del contenido web en forma de texto, así como para complementar el texto con objetos tales como imágenes, se escribe en forma de etiquetas, rodeadas por corchetes angulares (<,>). También puede describir hasta un cierto punto, la apariencia de un documento y es el lenguaje de publicación del *World Wide Web* (WWW), (21). Se utiliza en la realización de la aplicación *frontend* (Aplicación del lado del usuario) que se planea desarrollar ya que es el lenguaje que sirve de estructura para toda web.

CSS3

Hojas de Estilo en Cascada (del inglés *Cascading Style Sheets*) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (en-US). CSS describe cómo debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios, (22). Proporcionará a la aplicación *frontend* diversos estilos los cuales hacen más amigable y sencilla la interfaz gráfica para los usuarios.

TypeScript

Es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases. TypeScript es usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor, o extensiones para programas (23). Debido a que en el marco de trabajo se utiliza el *framework* Angular que está construido en TypeScript.

Python

Es un lenguaje interpretado de alto nivel de programación cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo, ejemplos: Instagram, Netflix, Spotify, Panda 3D, entre otros. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma, (24). Se seleccionó este lenguaje ya que es en el que se basa Django que es el *framework* para el desarrollo de la aplicación *backend* (Aplicación del lado del servidor) y además es el mismo en el que está implementado el Sistema de Gestión Académica XAUCE AKADEMOS.

UML

El Lenguaje Unificado de Modelado (UML) desempeña un rol importante no solo en el desarrollo de software, sino también en los sistemas que no tienen software en muchas industrias, ya que es una forma de mostrar visualmente el comportamiento y la estructura de un sistema o proceso, (25). Se utilizará este lenguaje de modelado ya que simplifica las complejidades y ayuda a resolver los problemas arquitectónicos constantemente, además aumenta la calidad del trabajo.

1.4.2 Herramienta

Las herramientas informáticas son programas, aplicaciones o simplemente instrucciones usadas para efectuar otras tareas de modo más sencillo. Cada herramienta se crea y diseña para una o varias funciones determinadas, por lo que existen diversos tipos de herramientas informáticas según el campo al que se dedique.

Suite de Visual Paradigm 16.3

Visual Paradigm es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación, (26). Esta selección es debido a las ventajas y facilidades que brinda esta herramienta para el modelado, además de mejorar la productividad y comprimir el desarrollo de software.

Nginx

Nginx es un servidor web ligero de alto rendimiento y de código abierto, licenciado bajo la Licencia BSD simplificada. Es multiplataforma, por lo que corre en sistemas tipo Unix (GNU/Linux, BSD, Solaris, Mac OS X, etc.) y Windows. Nginx consume menos recursos que sus principales competidores y es muy veloz con los contenidos estáticos. Es fácilmente escalable con recursos de hardware mínimos, (27). Su elección se debe a que es el servidor utilizado por el centro FORTES para el despliegue del proyecto XAUCE AKADEMOS.

PostgreSQL

PostgreSQL es un Sistema de Gestión de Bases de Datos (SGBD) relacional orientado a objetos y libre, publicado bajo la licencia PostgreSQL. Es conocido como el SGBD de código abierto más avanzado del mundo. Presenta estabilidad y confiabilidad, es extensible, multiplataforma, existen excelentes herramientas gráficas de diseño y administración de bases de datos y tiene buena escalabilidad ya que es capaz de ajustarse al número de CPU y a la cantidad de memoria disponible de forma óptima, soportando una mayor cantidad de peticiones simultáneas a la base de datos de forma correcta, (28). Es el SGBD utilizado por el centro de producción FORTES para gestionar todos los datos que almacena y con los que interactúa el sistema multiplataformas XAUCE AKADEMOS.

Visual Studio Code

Editor de código fuente independiente que se ejecuta en Windows, macOS y Linux. La elección principal para desarrolladores web y JavaScript, con extensiones para admitir casi cualquier lenguaje de programación, (29). Fue seleccionado este editor de texto debido a que el desarrollador tiene dominio sobre él, lo cual reduce el tiempo de aprendizaje y de desarrollo.

1.4.3 Marco de Trabajo

Angular

Angular es una plataforma y un marco de trabajo para construir aplicaciones de cliente de una sola página utilizando HTML y TypeScript. Angular está escrito en TypeScript. Implementa la funcionalidad principal y opcional como un conjunto de bibliotecas de TypeScript que importas en tus aplicaciones, (30). Es el *framework* utilizado para la construcción completa de la aplicación Frontend del Sistema multiplataformas XAUCE AKADEMOS y está definido su uso para todo módulo componente que se integre al mismo.

Django

Django es un marco web de alto nivel en Python que fomenta un desarrollo rápido y un diseño limpio y pragmático. Construido por desarrolladores experimentados, se encarga de gran parte de las molestias del desarrollo web, para que puedas centrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, (31). La aplicación *backend* de XAUCE AKADEMOS está construida por este *framework* por lo tanto cualquier componente o módulo que se integre al mismo deberá estar construido con Django.

1.5 Conclusiones parciales

- El estudio realizado permitió identificar los diferentes conceptos y definiciones más importantes para dar cumplimiento al problema de la investigación, las herramientas y tecnologías a tener en cuenta para el posterior desarrollo de la investigación.
- Como guía para el proceso de desarrollo se escogió la metodología ágil AUP UCI en su escenario 3, para el modelado de los artefactos, la herramienta CASE Visual Paradigm en su versión 16.3. Determinándose que la propuesta de solución será una aplicación web, para su desarrollo se seleccionó como lenguaje de programación Python, HTML5, TypeScript y CSS3. Para el almacenamiento de la información al gestor de bases de datos PostgreSQL, como *framework* de desarrollo a Django y como IDE fue seleccionado el Visual Studio Code.

CAPÍTULO II: Propuesta de Solución

En el capítulo se exponen las características de la propuesta de solución. Se presenta una descripción detallada de los requisitos funcionales y no funcionales, así como las técnicas para su obtención. Se describe la arquitectura, el diseño del sistema y los patrones utilizados.

2.1 Propuesta del Módulo de Ayuda en Línea

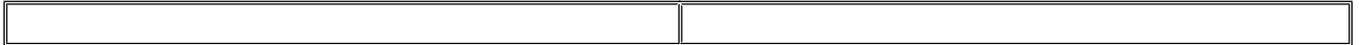
El presente trabajo propone la implementación de un módulo de ayuda en línea que provea las funcionalidades que dan respuestas a las necesidades de los usuarios. Dicha ayuda debe permitir al administrador la gestión (crear, modificar y eliminar) de las FAQ, que no son más que las preguntas que generan dudas con mayor frecuencia en los usuarios. Además contará con un sistema de ayuda interactiva para aquellos módulos que generan mayor incertidumbre en los usuarios. El administrador de la ayuda será el encargado de gestionar toda la información referente a la misma y dará soporte en cada una de las instancias necesitadas.

Personas relacionadas con el sistema

Se define como persona relacionada a la ayuda toda aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan en la misma. Además de aquellas que se encuentran involucradas en dichos procesos, pues participan en ellos pero no obtienen ningún resultado de valor.

Tabla 1 Personas que interactúan con la ayuda en línea

| Persona relacionada con la ayuda | Función |
|---|---|
| Usuario | Es quién va a consumir los servicios brindados por el módulo de ayuda en línea |
| Administrador | Es la persona facultada para toda modificación que se produzca para la gestión de la ayuda. |



2.2 Modelado del negocio

El modelado del negocio es una técnica para comprender los procesos de negocio de la organización. Con este flujo de trabajo pretendemos llegar a un mejor entendimiento de la organización donde se va a implantar el producto. (32)

Los objetivos del modelado de negocio son:

- Entender la estructura y la dinámica de la organización para la cual el sistema va a ser desarrollado (organización objetivo).
- Entender el problema actual en la organización objetivo e identificar sus potenciales y mejoras.
- Asegurar que los clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo.
- Derivar los requisitos del sistema necesarios para apoyar a la organización objetivo.

Para lograr estos objetivos, el modelo de negocio describe cómo desarrollar una visión de la nueva organización, basado en esta visión se definen procesos, roles y responsabilidades de la organización por medio de un modelo de Casos de Uso del negocio y un Modelo de Objetos del Negocio.

Complementario a estos modelos se desarrollan otras especificaciones

El modelado de negocio forma parte de una de las disciplinas definidas en la metodología AUP-UCI destinada a comprender los procesos de negocio de una organización. Debido al avanzado estado del sistema XAUCE AKADEMOS su negocio se encuentra modelado completamente. Por tanto esta disciplina no será desarrollada en esta tesis.

2.3 Requisitos del Software

Los requisitos para un sistema de software determinan lo que debe hacer el sistema y definen las restricciones en su funcionamiento e implementación, es decir, lo que el software debe hacer y bajo qué circunstancias debe hacerlo (33). La captura de requisitos es un paso fundamental para saber exactamente lo que debe hacer y las cualidades o propiedades que deben tener los componentes definidos. Para el desarrollo de la propuesta de solución se identificaron requisitos funcionales y no funcionales, siendo los primeros guiados mediante técnicas de obtención que permiten establecer una comunicación más específica entre el cliente y el equipo de desarrollo.

2.3.1 Definición de las técnicas de obtención de requisitos

La obtención de requerimientos es una de las cuatro actividades que define la Ingeniería de Requisitos, para ayudar a reconocer la importancia que tiene una especificación y administración adecuada de los requerimientos de los usuarios. La obtención de requisitos es el proceso donde los interesados en un sistema de software descubren, revelan y entienden sus requerimientos, evidenciando la necesidad de trabajar con el cliente debido a que sus requisitos abarcan las tareas que se necesitan para llevar a cabo el desarrollo del sistema (34). En la identificación de los requisitos de la propuesta de solución se utilizó la siguiente técnica:

- Entrevista: se utilizó con el fin de obtener diversas opiniones y recomendaciones en un intercambio mediante preguntas con los especialistas del centro FORTES, para esclarecer con precisión el funcionamiento del desarrollo del módulo de ayuda en línea.
- Desarrollo de Prototipos: Los prototipos suelen consistir en versiones reducidas, demos o conjuntos de pantallas (que no son totalmente operativos) de la aplicación pedida. Los prototipos de sistema permiten a los usuarios experimentar para ver cómo éste ayuda a su trabajo. Fomentan el desarrollo de ideas que desembocan en requerimientos. Además de permitir a los usuarios mejorar las especificaciones de requerimientos.

2.3.2 Definición de Requisitos Funcionales

Los requerimientos funcionales de un sistema describen lo que el sistema debe hacer. Dependen del tipo de software que se desarrolle, de los posibles usuarios y del enfoque general de la organización tomado al redactar los requerimientos (35). En la siguiente tabla se muestran los requerimientos funcionales definidos para la propuesta de solución.

Tabla 2 Listado de requisitos funcionales

| No. | Requisito |
|-------|--|
| RF1 | Gestionar Preguntas Frecuentes (FAQ's) |
| RF1.1 | Adicionar Pregunta |
| RF1.2 | Buscar Pregunta |

| | |
|-------|--|
| RF1.3 | Modificar Pregunta |
| RF1.4 | Eliminar Pregunta |
| RF2 | Reproducir ayuda interactiva para la gestión de la Versión de Asignatura |
| RF3 | Reproducir ayuda interactiva para la gestión de las Especialidades del MINED |
| RF4 | Reproducir ayuda interactiva para la gestión de las Versiones de plan de estudio |
| RF5 | Reproducir ayuda interactiva para la gestión de las Organizaciones de plan de estudio |
| RF6 | Reproducir ayuda interactiva para la gestión de los Rangos de calificación |
| RF7 | Reproducir ayuda interactiva para la gestión de los trámites de Registro |
| RF8 | Reproducir ayuda interactiva para la gestión de los trámites de Ubicaciones |
| RF9 | Reproducir ayuda interactiva para la gestión de los trámites de Matrícula |
| RF10 | Reproducir ayuda interactiva para la gestión de los trámites de Traslado |
| RF11 | Reproducir ayuda interactiva para la gestión de los trámites de Continuidad de estudio |
| RF12 | Reproducir ayuda interactiva para la gestión de los trámites de Cambio de educación |
| RF13 | Reproducir ayuda interactiva para la gestión de los trámites de Baja |
| RF14 | Reproducir ayuda interactiva para la gestión de los trámites de Cambio de Grupo |
| RF15 | Reproducir ayuda interactiva para la gestión de los trámites de Cambio de Período |
| RF16 | Reproducir ayuda interactiva para la gestión de los trámites de Ratificación |
| RF17 | Reproducir ayuda interactiva para la gestión de los trámites de Graduación |
| RF18 | Reproducir ayuda interactiva para la gestión de los Tipos de trámites |
| RF19 | Reproducir ayuda interactiva para la gestión de los Títulos |
| RF20 | Reproducir ayuda interactiva para la gestión de los Grupos de Certificación |
| RF21 | Reproducir ayuda interactiva para la gestión de los Trabajadores |

2.3.3 Definición de Requisitos No Funcionales

Son requerimientos que permiten definir las cualidades o propiedades que el software debe tener, con el objetivo de crear un producto final que sea fácil de utilizar, rápido y confiable. Los requisitos no funcionales imponen restricciones en el diseño o la implementación (36). En la Tabla 3 se muestran los requisitos no funcionales definidos para la propuesta de solución.

Tabla 3 Listado de requisitos no funcionales

| Usabilidad | |
|-----------------------|--|
| RNF1 | El módulo debe contar con una adecuada organización de las acciones permitiendo la fácil interacción con el usuario. |
| RNF2 | La ayuda debe ser de fácil manejo para los usuarios que tengan niveles básicos sobre la computación. |
| RNF3 | Debe facilitar, principalmente el ser manejado por usuarios que estén vinculados a los procesos que se llevan a cabo con el sistema XAUCE AKADEMOS. |
| Rendimiento | |
| RNF4 | La eficiencia de esta aplicación debe ser óptima en cuanto a la velocidad de procesamiento, disponibilidad, tiempo de respuesta y aprovechamiento de los recursos entre otros. |
| RNF5 | El tiempo de búsqueda de información debe ser en el menor tiempo posible. |
| Seguridad | |
| RNF6 | Proteger la información manejada por la ayuda de accesos no autorizados. |
| RNF7 | Mantener la integridad de la información, es decir que no se perderá durante su almacenamiento o transporte. |
| RNF8 | Mostrar opción de advertencia antes de borrar cualquier documento o información que pueda existir. |
| Disponibilidad | |
| RNF9 | La ayuda deberá estar disponible las 24 horas del día para todos los usuarios con permisos para utilizarla. |
| Apariencia | |
| RNF10 | Empleo de animaciones e imágenes sencillas, permitiendo la rapidez de la aplicación. |
| Hardware | |
| RNF11 | Las PC clientes deben contar con microprocesadores a velocidad |

| | |
|-----------------|---|
| | de 1.0GHz o superior; una memoria RAM de 512 o superior. |
| RNF12 | El servidor donde se efectuará el despliegue se precisa que posea los siguientes recursos: SO: Ubuntu Server 18.04 RAM: 16GB HD: 200GB |
| Software | |
| RNF13 | Las PC clientes deben contar con un navegador web |
| RNF14 | Para los servidores se debe contar con el servidor web NGINX |

2.3.4 Descripción de Requisitos Funcionales.

Tabla 4 Descripción de Requisitos Funcionales.

| Nº | Nombre | Descripción | Prioridad | Complejidad |
|-----|--|--|-----------|-------------|
| RF1 | Gestionar Preguntas Frecuentes (FAQ's) | El sistema debe mostrar las siguientes opciones: - Adicionar - Buscar - Modificar - Eliminar Cerrar | Alta | Media |
| RF2 | Reproducir ayuda interactiva para la gestión de la Versión de Asignatura | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF3 | Reproducir ayuda interactiva para la gestión de las Especialidades del MINED | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |

| | | | | |
|------|---|---|------|-------|
| RF4 | Reproducir ayuda interactiva para la gestión de las Versiones de plan de estudio | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF5 | Reproducir ayuda interactiva para la gestión de las Organizaciones de plan de estudio | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar. | Alta | Media |
| RF6 | Reproducir ayuda interactiva para la gestión de los Rangos de calificación | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF7 | Reproducir ayuda interactiva para la gestión de los trámites de Registro | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF8 | Reproducir ayuda interactiva para la gestión de los trámites de Ubicaciones | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF9 | Reproducir ayuda interactiva para la gestión de los trámites de Matrícula | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF10 | Reproducir ayuda interactiva para la gestión de los | El sistema debe activar la ayuda explicando detalladamente todos los pasos | Alta | Media |

| | | | | |
|------|--|--|------|-------|
| | trámites de Traslado | a seguir para hacer uso de este módulo. Cerrar | | |
| RF11 | Reproducir ayuda interactiva para la gestión de los trámites de Continuidad de estudio | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF12 | Reproducir ayuda interactiva para la gestión de los trámites de Cambio de educación | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF13 | Reproducir ayuda interactiva para la gestión de los trámites de Baja | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF14 | Reproducir ayuda interactiva para la gestión de los trámites de Cambio de Grupo | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF15 | Reproducir ayuda interactiva para la gestión de los trámites de Cambio de Período | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF16 | Reproducir ayuda interactiva para la gestión de los trámites de | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. | Alta | Media |

| | | | | |
|------|---|--|------|-------|
| | Ratificación | Cerrar | | |
| RF17 | Reproducir ayuda interactiva para la gestión de los trámites de Graduación | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF18 | Reproducir ayuda interactiva para la gestión de los Tipos de trámites | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF19 | Reproducir ayuda interactiva para la gestión de los Títulos | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF20 | Reproducir ayuda interactiva para la gestión de los Grupos de Certificación | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |
| RF21 | Reproducir ayuda interactiva para la gestión de los Trabajadores | El sistema debe activar la ayuda explicando detalladamente todos los pasos a seguir para hacer uso de este módulo. Cerrar | Alta | Media |

Para la representación de manera más sólida y por tanto más entendible se realizó la descripción detalladamente de cada Requisito Funcional, definiendo las acciones llevadas a cabo por el usuario para ejecutar cada funcionalidad. A continuación se muestra el ejemplo de la funcionalidad Adicionar Pregunta perteneciente al RF1.

Tabla 5 Descripción detallada del RF1.1 Adicionar Pregunta

| | | |
|--|--|---|
| Precondiciones | | Debe haber un usuario autenticado con permiso a la funcionalidad. |
| Flujo de eventos | | |
| Flujo básico Adicionar Pregunta Frecuente | | |
| 1. | Se selecciona la funcionalidad Preguntas /Adicionar. | |
| 1. | El sistema adiciona una nueva pregunta proporcionada por el administrador. | |
| 1. | Si selecciona una opción fuera de estas, se concluye el requisito. | |
| Pos-condiciones | | |
| 1. | Se adicionó la nueva pregunta a la lista de FAQ's. | |
| Flujos alternativos | | |
| N/A | | |
| Pos-condiciones | | |
| 1 | N/A | |
| Validaciones | | |
| 1 | | |
| Conceptos | Persona | Visibles en la interfaz: <ul style="list-style-type: none"> N/A Utilizados internamente: <ul style="list-style-type: none"> Usuario |
| | Registro_Trabajador | Visibles en la interfaz: <ul style="list-style-type: none"> N/A Utilizados internamente: <ul style="list-style-type: none"> trabajador responsabilidad |
| Requisitos especiales | N/A | |
| Asuntos pendientes | N/A | |
| Prototipo elemental de interfaz gráfica de usuario | | |
|  | | |

2.4 Arquitectura y Diseño de Software

La arquitectura de un software es la estructura u organización de un sistema que incluye los componentes de este, las propiedades visibles externas de esos componentes y las relaciones que existen entre ellos.(37) Es una representación que permite que un ingeniero de software analice la efectividad del diseño para cumplir con los requisitos establecidos, considere opciones arquitectónicas en una etapa en que aún resulta relativamente fácil hacer cambios al diseño y reduzca los riesgos asociados con la construcción del software.

Arquitectura Cliente-Servidor

- Cliente: Programa ejecutable que participa activamente en el establecimiento de las conexiones. Envía una petición al servidor y se queda esperando por una respuesta. Su tiempo de vida es finito una vez que son servidas sus solicitudes, termina el trabajo.
- Servidor: Es un programa que ofrece un servicio que se puede obtener en una red. Acepta la petición desde la red, realiza el servicio y devuelve el resultado al solicitante. Al ser posible implantarlo como aplicaciones de programas, puede ejecutarse en cualquier sistema donde exista TCP/IP y junto con otros programas de aplicación. El servidor comienza su ejecución antes de comenzar la interacción con el cliente (38).

El sistema se hospedará en el servidor web de aplicaciones Apache instalado en uno de los servidores de nodo central de la Universidad y se podrá acceder mediante la dirección URL <https://akademos.uci.cu>. El uso de esta arquitectura permite que los usuarios puedan acceder al sistema en cualquier dispositivo conectado a la red UCI logrando centralizar la información y facilitar el acceso a esta, sin necesidad de que todos tengan que utilizar el mismo sistema operativo.

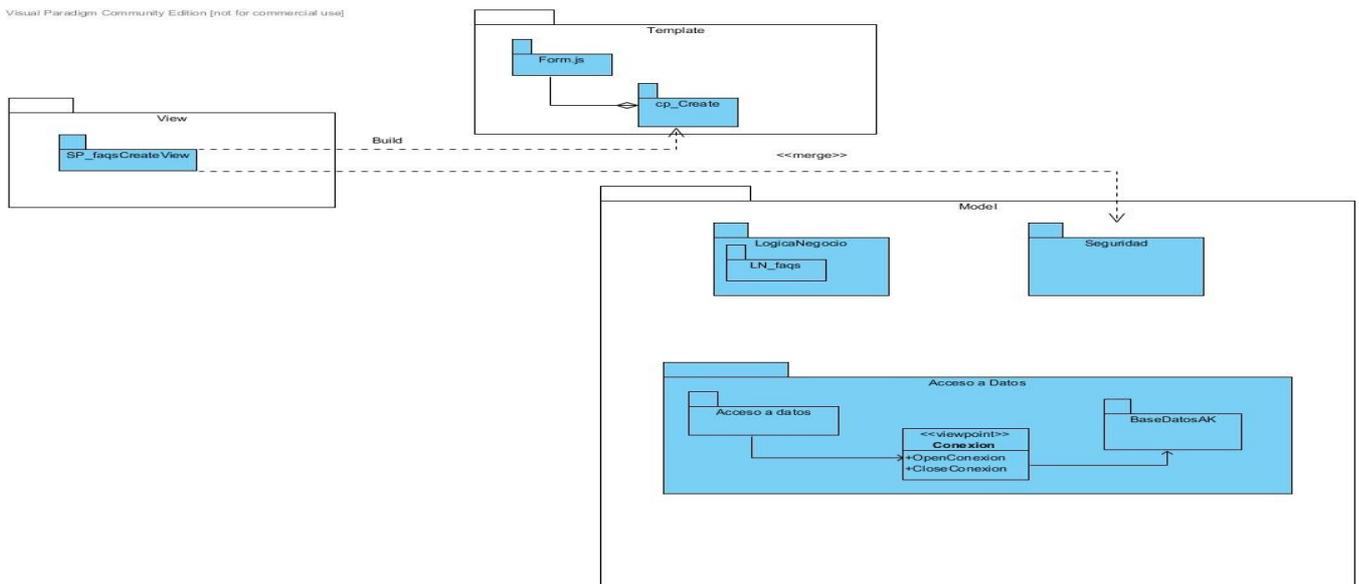
Patrón Arquitectónico

En el presente módulo, la clase “Controladora” es manejada por el mismo *framework* y la parte más importante se produce en los modelos, las plantillas y las vistas, Django usa este patrón con una nueva definición: Modelo-Vista-Plantilla (MVT). (31)

- M, significa “*Model*” (Modelo), la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: como acceder a estos, cómo validarlos, cuál es el comportamiento

- que tiene, y las relaciones entre los datos.
- T significa “*Template*” (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: cómo algunos elementos son mostrados sobre una página web u otro tipo de documento.
- V significa “*View*” (Vista), la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: se puede pensar en ella como un puente entre el Modelo y las Plantillas.

Ilustración 1 Diagrama de Paquetes correspondiente al Modelo-Vista-Plantilla



2.4.1 Patrones de diseño

“Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software.” En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares; ayudan al éxito del proyecto, pues permiten la reutilización de código, garantizan la robustez y extensibilidad del software. (Tedeschi, 2014)

Patrones GRASP

Los Patrones Generales de Software para Asignación de Responsabilidades (GRASP por sus siglas en inglés) describen los principios fundamentales de la asignación de responsabilidades a objetos,

expresados en formas de patrones. Ayudan a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Las responsabilidades están relacionadas con las obligaciones de un objeto en cuanto a su comportamiento (39). A continuación, se describen los patrones GRASP a utilizar en el desarrollo de la propuesta de solución:

- Experto: consiste en asignar una responsabilidad al experto en información, se asigna la responsabilidad a la clase que cuenta con la información necesaria para cumplirla. Se evidencia en las clases que extienden de la clase Entidad, las cuales son expertas en su propia información.

Ilustración 2 Patrón GRASP Experto

```
class Pregunta(models.Model):  
    titulo=models.CharField(max_length=30)  
    respuesta=models.TextField()
```

- Controlador: se basa en asignar la responsabilidad de todos los eventos realizados a una clase específica que constituye el único punto de entrada para cada evento. En la solución propuesta este patrón está evidenciado en la clase controladora.

Ilustración 3 Patrón GRASP Controlador

```
# Create your views here.  
def home(request):  
    return render(request, "base/base.html")
```

- Bajo Acoplamiento: el acoplamiento mide el grado en que una clase está conectada, tiene conocimiento o de alguna manera depende de otra. Este patrón consiste en asignar la responsabilidad de manera que el acoplamiento permanezca bajo. El bajo acoplamiento permite crear clases más independientes, más reutilizables, lo que implica mayor productividad.

Patrones GoF

Los patrones Banda de los Cuatro (*Gang-of-Four*) describen las formas comunes en que diferentes tipos de objetos, pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases y la formación de estructuras de mayor complejidad. Además, permiten crear grupos de objetos para

ayudarnos a realizar tareas complejas. Existen tres tipos de patrones: de creación, estructurales y de comportamiento. Los patrones de creación abstraen la forma en la que se crean los objetos, permitiendo tratar las clases a crear de forma genérica dejando para más tarde la decisión de qué clases crear o cómo crearlas (39). Los patrones GoF que se utilizarán en el desarrollo del módulo son:

- Mediator: define un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto.
- Patrones de creación: se encargan de la creación de instancias de los objetos. Abstraen la forma en que se crean los objetos, permitiendo tratar las clases a crear de forma genérica, dejando para después la decisión de que clase crear o cómo crearla.

2.5 Conclusiones Parciales.

Luego de describir las características que debe cumplir el módulo basado en la metodología AUP_UCI en el escenario 3 y de realizar el análisis y diseño se arrojaron los siguientes resultados:

- Se lograron identificar 21 requisitos funcionales y 14 no funcionales.
- El patrón arquitectónico utilizado es Modelo-Vista-Plantilla (MVT) usado en django
- Se construyeron 21 Descripción de Requisitos por Procesos (DRP) correspondientes a cada requisito funcional.

CAPÍTULO III: Validación de la Propuesta de Solución.

Un producto listo para ser entregado, requiere el completo desarrollo y validación de las funcionalidades previamente definidas. En el actual capítulo se describe la construcción de la solución propuesta teniendo en cuenta las técnicas de programación, estrategias de codificación y la integración con otras aplicaciones. Se define y aplica la estrategia de pruebas del producto, especificando las evaluaciones realizadas a la propuesta y los resultados.

3.1 Estándares de codificación

Los estándares de codificación son un conjunto de reglas a seguir por los desarrolladores con el objetivo de establecer un orden y un formato común en el código fuente del software en desarrollo. Son de vital importancia durante la etapa de construcción del software ya que permite que el personal del proyecto pueda entender de forma fácil el código, garantizándose la organización y estructura del código fuente (40). Para el desarrollo de la solución se utilizan los estándares de codificación establecidos por el grupo de arquitectura del Departamento de Desarrollo del Departamento de Informatización, con el propósito de estandarizar las nomenclaturas en la implementación del sistema y obtener un producto estable eficiente.

CamelCase

Es un estándar en el que los nombres de los identificadores están compuestos por una o más palabras juntas, iniciando cada palabra con letra mayúscula y el resto en minúscula, donde la primera letra del identificador no es mayúscula sino minúscula. Este estándar fue seleccionado para el nombramiento de las funcionalidades, la figura 3 ejemplifica el uso del CamelCase:

Ilustración 4 Ejemplo del estándar CamelCase

```
class CrearPregunta(CreateView):  
    model=Pregunta  
    template_name="GestionPreguntas/create_pregunta.html"  
    form_class=PreguntaForm  
    success_url = reverse_lazy('Home')
```

CSS

- Una declaración CSS siempre termina en punto y coma y los conjuntos de declaraciones se colocan entre llaves.
- Para hacer un código CSS legible, ponga una declaración en cada línea.
- Coloque la llave que cierra en una línea nueva.
- Para nombres compuestos de una propiedad utilice el guion (-).

En la figura 4 se muestra un ejemplo de este tipo reflejado en el código de la aplicación.

Ilustración 5 Ejemplo de CSS

```
309  caption {
310      padding-top: 0.5rem;
311      padding-bottom: 0.5rem;
312      color: #6c757d;
313      text-align: left;
314  }
```

HTML

- No colocar espacios entre la relación atributos-valor.
- Utilizar lowercase para el nombre de los atributos de las etiquetas.

En la figura 5 se muestra un ejemplo de este tipo reflejado en el código de la aplicación.

Ilustración 6 Ejemplo de HTML

```
</div>
<form class="for form-control form-row" method="post">

  <div class="card-body">
    {% csrf_token %}
    {% block forms %}

    {% endblock %}

  </div>
```

Django

- Un espacio a cada lado del operador binario [=, -, +=, ==,>, in,is not, and]
- La declaración de importación debe escribirse en líneas separadas.
- El lenguaje natural usa comillas dobles "..."
- ID de máquina Utilice comillas simples'...'

En la figura 6 se muestra un ejemplo de este tipo reflejado en el código de la aplicación.

Ilustración 7 Ejemplo de Django

```
class CrearPregunta(CreateView):  
    model=Pregunta  
    template_name="GestionPreguntas/create_pregunta.html"  
    form_class=PreguntaForm  
    success_url = reverse_lazy('Home')
```

3.2 Validación de requisitos

La validación de requisitos examina las descripciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto (41)

Criterios para validar los requisitos

Para validar los requisitos del sistema, según Pressman, se pueden chequear mediante un cuestionario guiado por un conjunto de interrogantes con el objetivo de descubrir la mayor cantidad de errores posibles. A continuación, se muestra las preguntas utilizadas, algunas fueron agregadas a las planteadas por Pressman por ser consideradas necesarias para la validación. (41)

Interrogantes para la validación de requisitos:

- a) ¿Está el requisito claramente definido? ¿Puede interpretarse mal?
- b) ¿Está identificado el origen del requisito (por ejemplo: persona, norma, documento)?
- c) ¿El planteamiento final del requisito ha sido contrastado con la fuente original?

- d) ¿El requisito está delimitado en términos cuantitativos?
- e) ¿Qué otros requisitos hacen referencia al requisito estudiado?
- f) ¿El requisito incumple alguna restricción definida?
- g) ¿El requisito es verificable? Si es así, ¿se podrá efectuar pruebas para verificar el requisito?
- h) ¿Se puede seguir el requisito en el modelo del sistema que se ha desarrollado?
- i) ¿Está el requisito asociado con los rendimientos del sistema o con su comportamiento?
- j) ¿El requisito está implícitamente definido?
- k) ¿El requisito es modificable?
- l) ¿El requisito está completo?
- m) ¿El requisito puede ser implementado?
- n) ¿El requisito puede ser probado?
- o) ¿El resultado de la evaluación de impacto es positivo?

Resultado de aplicar los criterios de validación

Luego de aplicar el conjunto de interrogantes para validar los requisitos definidos para el desarrollo del componente, se obtuvo el 100 % de aprobación por parte del cliente.

3.2.1 Técnicas de validación de requisitos

Con el objetivo de obtener una mayor calidad y demostrar que los requisitos definidos realmente describen al sistema que el cliente necesita; se utilizó la siguiente técnica para la validación de requisitos:

- Revisión técnica formal de requisitos: se realizan reuniones donde se examinan las descripciones del sistema buscando errores en el contenido o en la interpretación, ideas donde se necesitan aclaraciones, información incompleta, inconsistencias, requisitos contradictorios o requisitos imposibles o inalcanzables.

3.3 Pruebas de software

El desarrollo del software implica en sí, errores que pueden empezar a ocurrir desde el primer momento del proceso en el que los requerimientos pueden estar especificados de forma errónea, así como en los posteriores pasos del diseño e implementación. Es por esto, que se hace necesario contar con una actividad que garantice la calidad. Las pruebas son la actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados y una evaluación es hecha de algún aspecto del sistema o componente. (42)

Las pruebas son un conjunto de actividades para asegurar que el software implemente correctamente una función específica y que se corresponda con los requisitos del cliente (43).

3.3.1 Niveles de prueba

Los niveles de prueba son diferentes formas de verificar y validar un producto de software. El proceso de realización de las pruebas está compuesto por una serie de niveles entre los que se encuentran: el nivel de pruebas unitarias, el de pruebas de integración, el de pruebas del sistema y el de pruebas de aceptación (44).

Una vez implementado el módulo, fue sometido a los niveles de pruebas que se describen en lo adelante, los cuales ayudaron a la detección de errores existentes.

- Pruebas unitarias: son las pruebas que se realizan a los elementos más pequeños del componente. Son aplicables a los componentes representados en el modelo de implementación para verificar que los flujos de control y de datos estén cubiertos y que funcionen como se espera. Constituyen la primera fase de las pruebas que se aplican a un sistema y su objetivo es verificar que estos componentes estén correctamente codificados. (45)
- Pruebas de integración: son las pruebas que se realizan para asegurarse de que los componentes en el modelo de implementación funcionen correctamente cuando se combinan para ejecutar una funcionalidad. Estas pruebas descubren errores en las especificaciones de las interfaces de los paquetes. Verifican que las interfaces entre las entidades externas y las aplicaciones funcionan correctamente y que las especificaciones de diseño sean alcanzadas. Es el proceso de combinar y probar múltiples componentes juntos. (45)
- Pruebas de sistema: son las pruebas que se realizan cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados. En un ciclo iterativo estas pruebas ocurren más temprano, tan pronto como subconjuntos bien formados de comportamiento de caso de uso son implementados. El objetivo de esta prueba es asegurar la apropiada navegación dentro del componente, ingreso de datos, procesamiento y recuperación. (45)

Métodos de prueba

- Pruebas de caja blanca o estructural: la prueba de caja blanca, en ocasiones llamada prueba de caja de cristal, es un método de diseño que usa la estructura de control descrita como parte

del diseño al nivel de componentes para derivar los casos de prueba. Estas comprueban los caminos lógicos del software, proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Requieren del conocimiento de la estructura interna del sistema y son derivadas a partir de las especificaciones internas de diseño o el código. (45)

- Pruebas de caja negra o funcional: las pruebas de caja negra, también denominadas pruebas de comportamiento, se concentran en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Permiten encontrar funciones incorrectas o ausentes, errores de interfaz, rendimiento, inicialización y terminación, así como errores en estructuras de datos o en accesos a la base de datos externas. La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software y son realizadas sin tener conocimiento interno del sistema. (45). Dentro del método de caja negra se hace uso de la técnica Partición Equivalente por ser una de las más efectivas para examinar los valores válidos e inválidos de las entradas existentes en el sistema. La Partición Equivalente divide el dominio de entrada de un programa en clases de datos a partir de las cuales se derivan casos de prueba. (45)

3.3.2 Resultados obtenidos

Pruebas unitarias

Durante el desarrollo del componente de integración se llevaron a cabo pruebas unitarias para garantizar que el código implementado de una funcionalidad específica cumple con el resultado esperado. Estas pruebas fueron realizadas por el desarrollador empleando el método de caja blanca. A continuación se muestra un ejemplo de ellas.

Ilustración 8 Prueba unitaria de la clase TestUrls.

```
1 from django.urls import reverse
2 from rest_framework import status
3
4 from core.api.generics.test_base import BaseAPITest
5
6 from structure.api.serializers import srl_faqs
7 from structure.models import FAQs
8
9 class TestUrls (BaseAPITest):
10     def test_faqs (self):
11         self.faqs = FAQs.objects.all()
12         self.url = reverse('structure api:faqs')
```

Ilustración 9 Resultado mostrado en consola.

```
(venv) jj@jj-HP-255-G7-Notebook-PC:~/ak-mined-arch-test/server$ python manage.py test test
System check identified some issues:

WARNINGS:
auditlog.LogEntry.additional_data: (django_jsonfield_backport.W001) You are using Django 3.1 or newer, which already has a built-in JSONField.
    HINT: Use django.db.models.JSONField instead.

System check identified 1 issue (0 silenced).
.....S.....
-----
Ran 43 tests in 9.172s

OK (skipped=1)
(venv) jj@jj-HP-255-G7-Notebook-PC:~/ak-mined-arch-test/server$
```

Pruebas de integración

La necesidad de realizar las pruebas de integración viene dada por el hecho de que los módulos que forman un programa suelen fallar cuando trabajan de forma conjunta, aunque previamente se haya demostrado que funcionan correctamente de manera individual. Con el uso de estas pruebas se consigue ir formando el programa global a medida que se comprueba como los distintos componentes interactúan y se comunican libres de errores (46).

Durante la implementación de las funcionalidades se realizaron las pruebas de integración. Estas pruebas validaron la integración entre el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED y el Módulo de Ayuda en Línea.

Pruebas de Sistema

Las pruebas de sistema es una serie de diferentes pruebas cuyo propósito principal es ejercitar por completo el sistema basado en computadora. Aunque cada prueba tenga un propósito diferente, todo el sistema funciona para verificar que los elementos se hayan integrado de manera adecuada y que se realicen las funciones asignadas. (47)

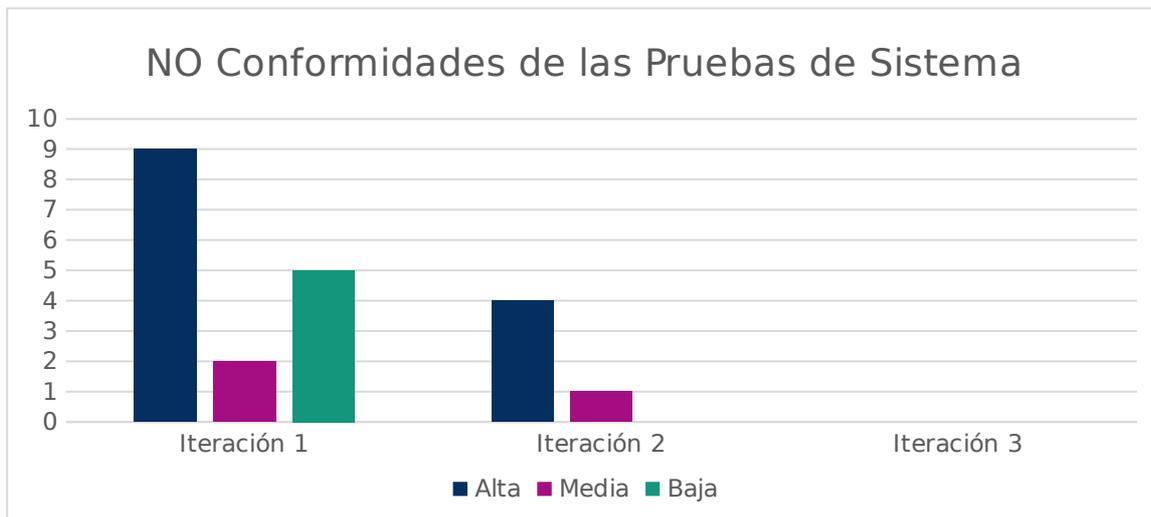
Este tipo de prueba se enfoca en validar la correcta implementación de las necesidades del cliente. La funcionalidad puede ser vinculada a los datos de entrada y de salida. Los datos de entrada serán ejecutados y mostrarán un resultado y dicho resultado será comparado con el resultado esperado (comportamiento). (47)

Para la realización de estas pruebas se utilizó el método de caja negra. A continuación, se muestra una tabla con la cantidad de iteraciones realizadas.

Tabla 6 No conformidades detectadas mediante las pruebas de sistema.

| Iteraciones | Cantidad de casos de prueba | No conformidades detectadas | | | |
|-------------|-----------------------------|-----------------------------|-------|------|-------|
| | | Alta | Media | Baja | Total |
| 1 | 21 | 9 | 2 | 5 | 16 |
| 2 | 21 | 4 | 1 | 0 | 5 |
| 3 | 21 | 0 | 0 | 0 | 0 |

Ilustración 10 Gráfica representativa de las no conformidades



Pruebas de aceptación

Las pruebas de aceptación constituyen una técnica para garantizar que los requisitos hayan sido cumplidos y que el sistema es realmente lo que el cliente necesita, además de asegurar su correcto funcionamiento. Estas son creadas a partir de los DRP y desde la perspectiva del cliente. El objetivo final es lograr que los requerimientos sean cumplidos y que el sistema sea aceptable. Tal es así, que una vez que todos los DRP hayan pasado sus pruebas de aceptación, se considera entonces terminado el sistema. Seguidamente se muestra el caso de prueba de aceptación del DRP “Adicionar pregunta”.

Tabla 7 Diseño de caso de prueba Adicionar pregunta

| Caso de prueba | | |
|---|------------------|--|
| Código de caso de prueba: 1 | | 1. Nombre DRP: Adicionar Pregunta |
| Nombre de la persona que realiza el caso de prueba: Luis David Senra Mirabal | | |
| Descripción de la prueba: revisar a través del sistema el correcto funcionamiento del RF <i>Adicionar Pregunta</i> . | | |
| Condiciones de ejecución: todos los campos deben ser rellenados. | | |
| Entrada/Pasos de ejecución | | Resultados esperados: |
| Acción: | Entrada: | |
| Se selecciona la opción: “Adicionar” | Datos requeridos | |
| El sistema debe mostrar un listado de las preguntas adicionadas. | | |
| Evaluación de prueba: Satisfactoria | | |

3.4 Conclusiones parciales

- La aplicación de las técnicas de validación de requisitos: revisión formal de los requisitos y generación de casos de prueba, permitieron obtener requerimientos entendibles y acordes a las necesidades del cliente, avalados por el acta de aceptación de requisitos obtenida al final del proceso.
- Estas técnicas facilitaron además el avance en la implementación del sistema informático y la retroalimentación con el cliente, obteniendo peticiones más exactas sobre las funcionalidades requeridas.
- La realización de pruebas unitarias y de aceptación propiciaron el uso de buenas prácticas profesionales, además de la obtención de errores de forma automática, contribuyendo a la celeridad del desarrollo del sistema informático.

Conclusiones

Luego de haber desarrollado el Módulo de Ayuda en Línea para el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED se concluye que:

- La investigación sobre el objeto de estudio y los métodos científicos utilizados permitieron identificar los conceptos y definiciones que respaldan la presente investigación. El análisis de las soluciones similares permitió conocer la existencia de herramientas diseñadas para la ayuda en línea y sirvió de guía para la construcción del módulo.
- La metodología seleccionada permitió generar los artefactos necesarios para obtener la aplicación, y satisfacer las necesidades del cliente como: Descripción de Requisitos por Procesos.
- Las pruebas realizadas permitieron verificar el correcto funcionamiento de las funcionalidades definidas en el sistema, el Módulo de Ayuda en Línea cumple con los requisitos definidos y la satisfacción del cliente.

Recomendaciones

Teniendo en cuenta los resultados obtenidos en la investigación se proponen las siguientes recomendaciones.

- Habilitar un medio de comunicación directa entre usuario y administrador.
- Desarrollar un chatbot que facilite respuestas a preguntas que no estén incluidas en las FAQ's.

Referencias bibliográficas

- 1- Ministerio de Educación de la República de Cuba. - Ministerio de Educación de la República de Cuba [en línea]. [sin fecha] [consultado el 8 de junio de 2022]. Disponible en: <https://www.mined.gob.cu/>
2. GARCÍA, Francisco; MOSCOSO, P. El sistema de ayuda al usuario de un catálogo en línea de acceso público: delimitación teórica y propuesta práctica. *Documentación de las Ciencias de la Información*, 2007, vol. 30, p. 187-199.
3. Métodos y técnicas de investigación científica. *gestiopolis* [en línea]. [sin fecha] [consultado el 7 de junio de 2022]. Disponible en: <https://www.gestiopolis.com/metodos-y-tecnicas-de-investigacion-cientifica/>
4. TORRES-MIRANDA, Teresa. En defensa del método histórico-lógico desde la Lógica como ciencia. *Revista Cubana de Educación Superior*, 2020, vol. 39, no 2.
5. AdLógica. [En línea] [consultado el 6 de junio de 2022]. <http://adlogica.com/glosario.html>.
6. Definición de Ayuda en Línea -Concepto y Significado. CarlosPes.com - Cursos, Libros, Videos y más Recursos Educativos [en línea]. [sin fecha] [consultado el 6 de junio de 2022]. Disponible en: http://www.carlospes.com/minidiccionario/ayuda_en_linea.php
7. MORE, Josh; STIEBER, Anthony; LIU, Chris. *Breaking Into Information Security: Crafting a Custom Career Path to Get the Job You Really Want*. Syngress, 2015.
8. CORBIN, Michelle. From online help to embedded user assistance. En *Annual Conference-Society for technical Communication*. UNKNOWN, 2003. p. 295-298.
9. GARCÍA, F.; GIL, A. Sistema de apoyo en Línea al usuario. *Universidad de Salamanca*, 2002
10. Zoho Desk | Software de servicio de atención al cliente con soporte contextual. Zoho [en línea]. [sin fecha] [consultado el 7 de junio de 2022]. Disponible en: <https://www.zoho.com/es-xl/desk/>
11. Jira | Software de seguimiento de proyectos e incidencias. *Atlassian* [en línea]. [sin fecha] [consultado el 7 de junio de 2022]. Disponible en: <https://www.atlassian.com/es/software/jira>

12. El Software de Help Desk para su empresa | ServiceTonic. ServiceTonic [en línea]. [sin fecha] [consultado el 8 de junio de 2022]. Disponible en: <https://www.servicetonic.com/es/>
13. Infomed, Portal de la Red de Salud de Cuba – Infomed, Portal de la Red de Salud de Cuba. *Infomed, Portal de la Red de Salud de Cuba – Infomed, Portal de la Red de Salud de Cuba* [en línea]. [sin fecha] [consultado el 7 de junio de 2022]. Disponible en: <https://www.sld.cu/>
14. Centro de Soporte | Universidad de las Ciencias Informáticas. Universidad de las Ciencias Informáticas [en línea]. [sin fecha] [consultado el 8 de junio de 2022]. Disponible en: <https://www.uci.cu/investigacion-y-desarrollo/centros-de-desarrollo/centro-de-soporte>
15. Picta | Dale Play! Picta | Dale Play! [en línea]. [sin fecha] [consultado el 8 de junio de 2022]. Disponible en: <https://www.picta.cu/about>
16. VELÁSQUEZ, Sandra Milena, et al. Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software. *Revista Cintex*, 2019, vol. 24, no 2, p. 13-23
17. PAZ, Julián Andrés Mera. Análisis del proceso de pruebas de calidad de software. *Ingeniería solidaria*, 2016, vol. 12, no 20, p. 163-176.
18. MONTERO, Bryan Molina; CEVALLOS, Harry Vite; CUESTA, Jefferson Dávila. Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espiraes revista multidisciplinaria de investigación*, 2018, vol. 2, no 17, p. 114-121.
19. GÓMEZ GUTIÉRREZ, Elizabeth, et al. Metodologías Ágiles para el Desarrollo de Proyectos. 2021.
20. SÁNCHEZ, Tamara Rodríguez. Metodología de desarrollo para la Actividad productiva de la UCI. *La Habana*, 2015.
21. SANDERS, Donald H. Informática presente y futuro. Editorial McGraw Hill, México, 1991.
22. HTML: Lenguaje de etiquetas de hipertexto | MDN. MDN Web Docs [en línea]. [sin fecha] [consultado el 10 de junio de 2022]. Disponible en: <https://developer.mozilla.org/es/docs/Web/HTML>
23. CSS | MDN. MDN Web Docs [en línea]. [sin fecha] [consultado el 10 de junio de 2022]. Disponible en: <https://developer.mozilla.org/es/docs/Web/CSS>
24. JavaScript With Syntax For Types. *TypeScript: JavaScript With Syntax For Types*. [en línea]. [sin fecha] [consultado el 10 de junio de 2022]. Disponible en: <https://www.typescriptlang.org/>

25. Welcome to Python.org. *Python.org* [en línea]. [sin fecha] [consultado el 10 de junio de 2022]. Disponible en: <https://www.python.org/>
26. JACOBSON, Lvar; BOOCH, James Rumbaugh Grady. The unified modeling language reference manual. 2021.
27. Ideal Modeling & Diagramming Tool for Agile Team Collaboration. Ideal Modeling & Diagramming Tool for Agile Team Collaboration [en línea]. [sin fecha] [consultado el 10 de junio de 2022]. Disponible en: <https://www.visual-paradigm.com/>
28. Advanced Load Balancer, Web Server, & Reverse Proxy - NGINX. NGINX [en línea]. [sin fecha] [consultado el 10 de junio de 2022]. Disponible en: <https://www.nginx.com/>.
29. PostgreSQL. PostgreSQL [en línea]. [sin fecha] [consultado el 10 de junio de 2022]. Disponible en: <https://www.postgresql.com>
30. MICROSOFT. Visual Studio Code - Code Editing. Redefined. Visual Studio Code - Code Editing. Redefined [en línea]. 3 de noviembre de 2021 [consultado el 10 de junio de 2022]. Disponible en: <https://code.visualstudio.com/>.
31. Angular. Angular [en línea]. [sin fecha] [consultado el 10 de junio de 2022]. Disponible en: <https://angular.io/>
32. The web framework for perfectionists with deadlines | Django. The web framework for perfectionists with deadlines | Django [en línea]. [sin fecha] [consultado el 10 de junio de 2022]. Disponible en: <https://www.djangoproject.com/>
33. Modelado del negocio. *CIDECAME UAEH* [en línea]. [sin fecha] [consultado el 24 - 09 de 2022]. Disponible en: http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro10/351_modelado_del_negocio.html
34. BIZAGI. *Guía de usuario aplica para la Versión 10.3.0 de Bizagi BMP Suite*. [En línea]. 2013. [Consultado: 24 de 09 de 2016]. Disponible en: <http://help.bizagi.com/bpmsuite>
35. PRESSMAN, R. S. *Ingeniería de Software, un enfoque práctico*, Séptima Edición. New York, McGraw Hill Companies, 2007. 264p.
36. PRESSMAN, R. S. *Ingeniería del Software*. Mexico, Mcgraw-Hill, 2005.
37. Keepcoding. [En línea] 2022. [Citado el: 20 de 06 de 2022.] <https://keepcoding.io/blog/ques-arquitectura-software/>.
38. Redes de computadores. [En línea] [Citado el: 20 de 06 de 2022.] <http://profesores.elo.utfsm.cl/~agv/elo322/1s16/projects/reports/Proyecto%20Cliente%20-%20Servidor.pdf>.
39. LARMAN, C. *UML Y PATRONES, Una Introducción al análisis y diseño orientado a objetos y al proceso unificado*. Madrid, Pearson Educación, 2003.

40. MICROSOFT. *Revisiones de código y estándares de codificación*. [En línea]. 2014. [Consultado: 19 de 10 de 2022]. Disponible en: <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>
41. PRESSMAN, R. S. *Ingeniería de software: Un enfoque práctico, Quinta edición*. Madrid, McGraw-Hill, 2002.
42. CARBALLO MUÑOZ, Lenna; BARRIENTOS NÚÑEZ, Ivette. Las causas del cambio en los requerimientos de software. *Revista Cubana de Ciencias Informáticas*, 2020, vol. 14, no 2, p. 131-144.
43. SERNA, Edgar; MARTÍNEZ, Raquel; TAMAYO, Paula. Una revisión a la realidad de la automatización de las pruebas del software. *Computación y Sistemas*, 2019, vol. 23, no 1, p. 169-183.
44. MARIN DIAZ, Aymara; TRUJILLO CASAÑOLA, Yaimí; BUEDO HIDALGO, Denys. Estrategia de pruebas para organizaciones desarrolladoras de software. *Revista Cubana de Ciencias Informáticas*, 2020, vol. 14, no 3, p. 83-104.
45. SOMMERVILLE, Ian. *Ingeniería del software*. Pearson educación, 2005.
46. VENCE Pérez, Jose María. *Plan de pruebas de integración*. Madrid: s.n., 2010.
47. CHIU, CINDY CAMPOS. LAS PRUEBAS EN EL DESARROLLO DE SOFTWARE. *UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO*. México: sn, 2015.

Anexos

Documentos que complementan el cuerpo del trabajo, pero que no son indispensables para entenderlo. Todos deben ser referenciados en el cuerpo del documento.