



Facultad 4

Título: Módulo Trámites Docentes 1.1 del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED

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

Autora: Yalé Martínez Montero

Tutores: Ing. Sandy Nuñez Padrón

Ms.C. Yadira Ramírez Rodríguez

Ing. Daniela Hernández Darías

La Habana, diciembre de 2022

Año 64 de la Revolución

DECLARACIÓN DE AUTORÍA

La autora del trabajo de diploma con título “**Módulo Trámites Docentes 1.1 del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED**” concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declaran como únicos autores de su contenido. Para que así conste firman la presente a los <2> días del mes de diciembre del año 2022.

Yaylé Martínez Montero



Firma del Autor

Ing. Sandy Nuñez Padrón



Firma del Tutor

Ms.C. Yadira Ramírez Rodríguez



Firma del Tutor

Daniela Hernández Darías



Firma del Tutor

AGRADECIMIENTOS

A mi mamá por siempre estar para mí, por su amor, su dedicación y paciencia.

A mi papá que desde lejos siempre estuvo presente en esta importante etapa.

A mi Neno Cristian por ser un pilar importante durante estos 5 años, por ser ese compañero ejemplar y por aguantarme mis malcriadeces.

A mis grandes amores de mi vida versión amigos Lilianna y Félix por ser mi mejor compañía, por siempre confiar en mí y darme la fuerza que necesite en momentos difíciles.

A mis tutores por guiarme en el desarrollo de esta investigación, por su paciencia y dedicación.

En general agradezco a toda mi familia y amigos.

DEDICATORIA

A mis padres, especialmente a mi mamá por ser mi mayor ejemplo y apoyarme en todo momento. Sin ustedes no hubiese sido posible, este logro es de los tres.

RESUMEN

El centro FORTES tiene entre sus objetivos el desarrollo del Sistema de Gestión Académica XAUCE AKDEMOS para el MINED, a través del cual se gestionan los procesos académicos de una institución. La presente investigación tiene como propósito desarrollar uno de los módulos de este sistema "Trámite Docente". Para dar solución a la propuesta se usa la metodología de desarrollo de software Proceso Unificado Ágil en su variación para la Universidad de las Ciencias Informáticas (AUP-UCI), así como las herramientas, tecnologías y lenguajes utilizados por el centro para este proyecto, como el framework angular, el gestor de base de datos PostgreSQL y el lenguaje de programación Python. Con el fin de satisfacer las necesidades del cliente, se evalúa la calidad del producto mediante varios métodos teniendo como resultado el módulo Trámite Docente 1.1 para el Sistema de Gestión Académica XAUCE AKDEMOS para el MINED, posibilitando así una mejor utilización del producto referente a los trámites docentes del país.

Palabras claves: procesos académicos, trámite docente.

ABSTRACT

The FORTES center has among its objectives the development of the Academic Management System XAUCE AKDEMOS for MINED, through which the academic processes of an institution are managed. The purpose of this research is to develop one of the modules of this system "Teaching Procedures". In order to provide a solution to the proposal, the Agile Unified Process software development methodology is used in its variation for the University of Computer Sciences (AUP-UCI), as well as the tools, technologies and languages used by the center for this project, such as the Angular framework, the PostgreSQL database manager and the Python programming language. In order to meet the client's needs, the quality of the product is evaluated by several methods, resulting in the Trámite Docente 1.1 module for the Academic Management System XAUCE AKDEMOS for MINED, thus enabling a better use of the product regarding the country's teaching procedures.

Keywords: academic processes, teaching procedure.

Tabla de contenido

INTRODUCCIÓN.....	1
Capítulo 1. Fundamentación teórica	6
1.1 Conceptos asociados.....	6
1.2 Estudios similares	7
1.3 Xauce Akademos	8
1.4 Metodologías de desarrollo de software.....	8
1.5 Herramientas y tecnologías a utilizar.....	10
1.6 Conclusiones parciales	13
Capítulo 2: Propuesta de solución	15
2.1 Descripción de la propuesta de solución	15
2.2 Disciplina Modelado de Negocio	15
2.3 Disciplina Requisitos	19
2.4 Disciplina Análisis y diseño	23
2.4.1 Arquitectura de la solución.....	24
2.5 Conclusiones parciales	28
Capítulo 3: Implementación y pruebas.....	1
3.1 Disciplina implementación.....	1
3.3.1 Diagrama de despliegue	1
3.1.2 Diagrama de componente.....	2
3.2 Estrategia de prueba.....	2
3.3 Pruebas de Software.....	4
3.3.1 Pruebas unitarias	4
3.4 Disciplina Pruebas Internas.....	9
3.5 Disciplina Aceptación	13
3.6 Conclusiones parciales	13
Conclusiones generales.....	15
Recomendaciones	16
Referencias bibliográficas	17
Anexos	22
Anexo 1 : Lista de chequeo para la prueba de usabilidad	22
Anexo 2: Lista de chequeo de la prueba de seguridad.....	27
Anexo 3: Prototipo de interfaz. Registrar trámite de matrícula.....	29
Anexo 4: Prototipo de interfaz. Ver detalles de trámite de traslado.	30
Anexo 5: Prototipo de interfaz. Registrar trámite de ubicación por registro	30
Anexo 6: Prototipo de interfaz. Listar estudiantes	31

Indice de ilustraciones

Ilustración 0.1 Modelo conceptual.....	16
Ilustración 0.2 Diagrama de análisis del diseño. Registrar trámite de matrícula.....	25
Ilustración 0.3 fragmento de código referente al patrón experto.....	26
Ilustración 0.4 fragmento de código referente al patrón bajo acoplamiento.....	26
Ilustración 0.5 fragmento de código referente al patrón alta cohesión.....	27
Ilustración 0.6 fragmento de código referente al patrón creador	27
Ilustración 0.7 fragmento de código referente al patrón estructural.....	27
Ilustración 0.8: fragmento de código referente al patrón de comportamiento	28
Ilustración 0.1 Diagrama de despliegue	2
Ilustración 0.2 Diagrama de componente.....	2
Ilustración 0.3 Fragmento Del Código Validar Formulario	6
Ilustración 0.4 Grafo De Flujo Asociado Al Método Validar Formulario	7
Ilustración 0.5 Resultados De La Pruebas De Rendimiento.....	10
Ilustración 0.6 No Conformidades Identificadas Por Iteraciones De Prueba.....	11
Ilustración 0.7 No Conformidades Identificadas Por Iteraciones De Prueba De Usabilidad	12
Ilustración 0.8 No Conformidades Identificadas Por Iteraciones De Prueba De Seguridad	13

Indice de tablas

Tabla 1. Reglas del negocio. Fuente: Elaboración propia	17
Tabla 2 Requisitos Funcionales. Elaboración Propia	20
Tabla 3. Estrategia De Prueba: Elaboración Propia	3
Tabla 4. Diseño De Caso De Prueba Para El Camino 1: Elaboración Propia.....	8

INTRODUCCIÓN

En la actualidad, las Tecnologías de la Información y la Comunicación (TIC) tienen un gran impacto en las distintas esferas de la sociedad. Sus usos en la sociedad han influido en gran medida en la manera de comunicación, convivencia y trabajo de las personas, sobre todo, ha dado un desvío en la forma en que se conciben nuestras vidas. Las TIC (Tecnologías de la información y la comunicación) son el conjunto de avances tecnológicos que nos proporcionan la informática, las telecomunicaciones y las tecnologías audiovisuales. Esto ha dado pie a nuevas formas de trabajar de comunicarse, divertirse, aprender y enseñar. En definitiva, la cultura en nuestro siglo está asociada a las tecnologías de la información. Están presentes en casi todas nuestras actuaciones y debemos convivir con ellas. (Monzó, 2016)

Por otro lado, se genera una dependencia tecnológica que es cada vez más patente, ya que la intención de mejorar nuestra calidad de vida y nuestros conocimientos da lugar a la formación de un pensamiento dependiente, confiando en la efectividad de la tecnología. Además, democratizan su uso, divulgan la cultura, y ofrecen información para que las personas estén informadas, tienen la capacidad de adormecer y movilizar grupos sociales, por medio de esta comunicación de masas, en las que se concretan las diferentes corrientes de opinión a través de personajes mediáticos y visibles dentro de nuestra sociedad. (Monzó, 2016)

La incorporación de las tecnologías de la información y comunicación (TIC) en el aula es un proceso que se está incrementando de manera acelerada a nivel mundial, es una expresión global de lo educativo. Ante esta situación es necesario tener presente que su incorporación no se limita al problema de contar con las herramientas que conforman estas tecnologías: equipos y programas de cómputo, sino que lo más importante es construir un uso educativo y, en estricto sentido, didáctico de las mismas. (Díaz-Barriga, 2013)

Cuba en los últimos 10 años ha potenciado el uso masivo de las TICs en el área de la Educación contribuyendo a la alfabetización digital de los estudiantes y de la sociedad en general. En este sentido, se han creados espacios donde investigadores y docentes exponen sus experiencias en el uso de las tecnologías (Molina., 2018). El sistema educacional del país avanza a un paso acelerado, por lo que es necesario almacenar todos los datos relacionados con el mismo, los cuales facilitan y mejoran los procesos formativos que imparten las instituciones educativas.

El Ministerio de Educación de Cuba (MINED) es el encargado de organizar y dirigir el sistema educativo de la República de Cuba. Si bien el ministerio existió durante muchos años, no fue hasta el triunfo de la revolución cubana en 1959, con la campaña de

alfabetización y posteriormente el incremento de las escuelas públicas, que el MINED cobró verdaderamente fuerza y comenzó a desempeñar un papel decisivo en el país (Ministerio de Educación Superior, s.f.).

La Universidad de Ciencias Informática (UCI), institución dedicada al desarrollo de software y servicios informáticos, posee varios centros productivos, entre los que se destaca el Centro de Tecnología para la Formación (FORTES), enfocado principalmente al desarrollo de sistemas que apoyen el proceso de enseñanza-aprendizaje. Uno de los proyectos que se lleva a cabo es el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED, en función del propósito manifestado anteriormente, fomentando la preparación de los recursos educativos y trazando líneas de investigación que aborden las diversas aristas para la gestión de la educación.

XAUCE AKADEMOS es una herramienta que contribuye al perfeccionamiento de los procesos académicos de una institución. Todos los roles del proceso educativo están involucrados en la solución, por lo que se permitirá el acceso a la información de forma segura a todos los niveles, facilitando la toma de decisiones.

Debido al cúmulo de información que gestiona, las incidencias detectadas y sugerencias de los clientes, se ha constatado que existen equivocaciones frecuentes por parte del usuario en la realización de determinadas acciones en el módulo Trámites Docentes y se necesitan cubrir nuevas necesidades del cliente. El mismo tiene una interfaz gráfica que no es adecuada para usuarios con determinadas características y discapacidades. Lo anterior, puede comprometer el correcto uso del sistema, e inclusive, el desuso del mismo en varios centros del país, lo que entraña una dispersión en la información dada la heterogeneidad que vislumbra dicho proceso.

Estas debilidades podrían traer como consecuencia vulnerabilidades tales como:

- Demoras en distintas etapas del proceso, como por ejemplo la matrícula y el traslado de un estudiante.
- Algunas funcionalidades similares no se realizan de igual forma, lo que dificulta la experiencia del usuario.
- A algunos usuarios se les hace difícil el uso del sistema y/o tienen poca confianza en el mismo, debido a que una parte importante de ellos presenta alguna discapacidad visual y/o carece de dominio de sistemas informáticos.
- Algunos datos estadísticos de interés gubernamental y social pueden verse afectados, por lo cual se puede comprometer la transparencia del proceso a partir de irregularidades en algunas de sus salidas.

En consecuencia, se plantea el siguiente **problema a resolver** ¿Cómo corregir las deficiencias manifestadas por los usuarios sobre el módulo Trámites Docentes del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED?

Para dar solución al problema a resolver se define como **objetivo general**: Desarrollar el módulo Trámites Docentes 1.1 del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED para corregir las deficiencias manifestadas por los usuarios.

La presente investigación precisa como **objeto de estudio**: la gestión de trámites docentes en sistemas de gestión académica y como **campo de acción**: la gestión de trámites docentes en el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED.

Preguntas científicas

- ¿Cómo se presenta la gestión de trámites docentes en sistemas de gestión académica, en la literatura científica?
- ¿Cómo desarrollar el módulo Trámites Docentes 1.1 del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED en función de la aplicación de una metodología de desarrollo de software, las herramientas y las tecnologías empleadas?
- ¿Cómo obtener una solución que satisfaga los índices de calidad del módulo Trámites Docentes 1.1 del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED?

Tareas de investigación

1. Análisis de los diferentes tópicos de la investigación, que comprende la gestión de trámites docentes en un sistema de gestión académica, así como las tecnologías empleadas.
2. Análisis, diseño e implementación del módulo Trámites Docentes 1.1 del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED a partir de la aplicación de una metodología de desarrollo de software.
3. Validación del módulo Trámites Docentes 1.1 del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED mediante el uso de pruebas.

Durante el desarrollo del presente trabajo de diploma, se utilizaron los siguientes métodos de investigación:

Métodos teóricos:

- **Analítico-sintético**: para descomponer el problema de investigación en

elementos por separado, de esta forma se puede profundizar en el estudio de cada elemento, para luego sintetizarlos en la solución de la propuesta.

- **Histórico-lógico:** Caracteriza al objeto en sus aspectos más externos, a través de la evolución y desarrollo histórico del mismo. Reproduce en el plano teórico la esencia del objeto de estudio. El empleo de este método posibilita realizar el estudio del estado del arte, investigar sobre los sistemas de gestión académica obteniendo así, sus características y elementos fundamentales, para un mejor entendimiento sobre el desarrollo del módulo que se quiere actualizar.

Métodos empíricos:

El método **observación científica** para estudiar más de cerca el objeto de la investigación, las acciones, causas y consecuencias. Permite tener más información acerca de los trámites docentes como de la gestión académica

Técnicas y procedimientos:

Técnicas de obtención de información y recolección de datos:

Revisión bibliográfica y documental: para identificar las premisas teóricas de los temas que se abordan en este estudio, se realizó un análisis de la bibliografía nacional e internacional disponible para las soluciones homólogas.

El presente documento está estructurado en tres capítulos cuyos contenidos son:

Capítulo 1. Fundamentación Teórica: en este capítulo se evidencian los términos asociados al tema de investigación. Se analizan las herramientas y metodologías de desarrollo de software a utilizar. Además, se realiza un análisis de las soluciones homólogas existentes mediante las cuales se pudo establecer un punto de partida para posteriormente desarrollar el módulo Trámites Docentes 1.1 del Sistema de Gestión Académica XAUCE AKADEMOS.

Capítulo 2. Propuesta de Solución: En este capítulo se describe la solución propuesta, se explica el proceso de desarrollo de software que conlleva a la solución. Además, se presentan diferentes elementos de la ingeniería de requisitos desarrollada para el sistema propuesto. Se documentan las disciplinas de requisito y de análisis y diseño de la solución propuesta, teniendo en cuenta la metodología de desarrollo de software seleccionada, lo cual permite la generación de diferentes artefactos.

Capítulo 3. Implantación y pruebas: En el capítulo se definirán los tipos y niveles de pruebas con el fin de liberar un módulo de calidad que cumpla los objetivos propuestos. Además, se mostrarán los resultados de las pruebas realizadas.

Capítulo 1. Fundamentación teórica

En el presente capítulo se presenta el estudio de los principales conceptos referentes al tema trámites docentes. Además, se analizan sistemas para gestionar dichos trámites a partir de un estudio de sistemas homólogos, las principales metodologías de desarrollo de software, así como las tecnologías y herramientas para el desarrollo de sistemas de gestión.

1.1 Conceptos asociados

Gestión Académica es aquel ámbito de acción orientada a facilitar y mejorar los procesos formativos que imparten las instituciones de educación superior. Esto incluye la definición, implementación y seguimiento de procesos de mejoramiento continuo que les permitan responder y anteponerse a las necesidades formativas de sus estudiantes, a las exigencias del entorno, del mercado laboral y la política pública. (Comisión Nacional de Acreditación CNA-Chile, 2018)

Trámite docente

Cada uno de los estados o diligencias necesarios para resolver un asunto es lo que se conoce como **trámite** (WordReference.com, s.f.) e. Un **docente** es aquel individuo que se dedica a enseñar o que realiza acciones referentes a la enseñanza. (WordReference.com, s.f.)

Teniendo en cuenta los últimos dos conceptos mencionados se puede llegar a la conclusión de que un trámite docente, de acuerdo a la presente investigación, es el conjunto de actividades que se realizan para gestionar el traslado, la matrícula, la baja de un estudiante en una institución educacional determinada.

La información es el conjunto de datos que configuran un mensaje que emite un emisor y que se pretende llegue al receptor para que quede informado. Siempre se lleva a cabo a través de un canal que es el que uno a ambos interlocutores han elegido. (Rosario Peiró, 2020)

Un sistema de gestión es una herramienta que permite controlar, planificar, organizar y, hasta cierto punto, automatizar las tareas de una organización. Su objetivo es unificar en un único software todas las operaciones de la compañía con el fin de facilitar la toma de decisiones y el análisis de los datos. (Ekon, 2021)

Por tanto, los sistemas de gestión académica junto con la utilización de las tecnologías han posibilitado informatizar todos los procesos docentes. Estos sistemas constituyen una poderosa herramienta de trabajo permitiendo hacer más eficientes dichos procesos y perfeccionar sus recursos.

1.2 Estudios similares

AKADEMOS, UN SISTEMA AUTOMATIZADO PARA LA GESTIÓN ACADÉMICA

AkademOS, es sistema de gestión universitaria posibilita un mejor control e incidencia en el aprendizaje de los estudiantes, pues con la información que los profesores introducen en tiempo real. Además, sus reportes pueden servir para efectuar investigaciones sociales importantes, dada la variedad en términos de proveniencia geográfica, social y académica de la masa estudiantil en la UCI. (Grisel Infante Costa, 2007)

- Utiliza SQL Server 2000 para el almacenamiento de los datos.
- IIS (Internet Information Services) como servidor Web.
- Servicios WEB XML para el intercambio con otras aplicaciones

SISTEMA DE GESTIÓN ACADÉMICA DE POSTGRADO VERSIÓN 2.0

Se describen los antecedentes que presenta el proceso de gestión académica de postgrado en la Universidad de las Ciencias Informáticas (UCI) y se realiza un análisis concerniente a los problemas que presenta el Sistema de Gestión Académica de Postgrado versión 2.0 (Universidad de las Ciencias Informáticas, 2014). Este sistema brinda información sobre los trámites para la formación académica de postgrado, como las inscripciones y las bajas de los estudiantes de las actividades.

- Metodología Desarrollo Ágil con Calidad
- Arquitectura cliente-servidor
- Patrón arquitectónico: modelo-vista-controlador
- Utiliza los patrones GOF

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA LA GESTIÓN ACADÉMICA DE UN INSTITUTO SUPERIO TECNOLÓGICO

Es un sistema que gestiona la información académica de los Institutos Superiores Tecnológicos. Asegura el manejo eficiente y contribuye a incrementar la calidad de los servicios brindados a los alumnos (Guevara, 2011). Este sistema presenta el módulo Alumnos el cual se encarga de administrar la información relacionada con la matrícula y el traslado de los alumnos.

- Lenguaje de programación Java Edición estándar 1.6.0
- Entorno de desarrollo NetBeans IDE 6.9.1
- Base de Datos PostgreSQL 8.4.0
- Metodología Proceso Unificado Rational (RUP).

El análisis de las soluciones homólogas permitió entender que ninguna constituye una solución factible al problema de investigación. En cuanto a las metodologías empleadas y herramientas que utilizan no se corresponden con las utilizadas en el del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED. A pesar de que en los sistemas estudiados no tienen un módulo Trámite Docente, cuentan con funcionalidades con las que debe contar la actualización del módulo por lo que se toman como referencia, entre las que se encuentran los registros de los estudiantes, los trámites de traslados y los trámites de baja.

1.3 Xauce Akademos

El objetivo principal de XAUCE AKADEMOS es gestionar información de los procesos académicos para contribuir a la toma de decisiones a partir del registro, control y seguimiento del tránsito del estudiante por los subsistemas del Sistema Nacional de Educación que atiende el MINED. Para ello, gestiona la estructura del Sistema Nacional de Educación, los inmuebles, los centros, los planes de estudio, el calendario escolar, los procesos de trámite docente, los medios tecnológicos, las actividades laborales y productivas y el abastecimiento estudiantil.

El módulo Trámite Docente actualmente cuenta con las funcionalidades de gestionar el cambio de grupo, el cambio de periodo y la ratificación de estudio de un estudiante, así como. Esta investigación se basa en actualización del módulo Trámite Docente, garantizando más gestiones de trámites docentes de un estudiante, dependiendo de las necesidades del cliente, por ejemplo, tramitar una baja, registrar una matrícula, registrar una modificación por cambio de estudio, tramitar un traslado y/o una baja.

1.4 Metodologías de desarrollo de software

Las metodologías de desarrollo de software se utilizan en el ámbito de la programación, con el objetivo de trabajar en equipo de manera organizada. Estas metodologías han ido evolucionando a lo largo del tiempo, pasando de ser un mero trámite de organización a ser una base importantísima a la hora de desarrollar software de una manera productiva y eficaz. (Santander Universidades , 2020)

Las metodologías de desarrollo de software son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático. El objetivo de las distintas metodologías es el de intentar organizar los equipos de trabajo para que estos desarrollen las funciones de un programa de la mejor manera posible. (Santander Universidades , 2020)

La metodología hace referencia al conjunto de procedimientos racionales utilizados para alcanzar un objetivo que requiera habilidades y conocimientos específicos. Es una de

las etapas específicas de un trabajo o proyecto que parte de una posición teórica y conlleva a una selección de técnicas concretas o métodos acerca del procedimiento para el cumplimiento de los objetivos. Es el conjunto de métodos que se utilizan en una determinada actividad con el fin de formalizarla y optimizarla. Determina los pasos a seguir y cómo realizarlos para finalizar una tarea. (Maida, Esteban Gabriel & Pacienza, Julián, 2015)

La autora de la presente investigación propone trabajar con lo planteado por (Maida, Esteban Gabriel & Pacienza, Julián, 2015) respecto a la metodología de desarrollo de software.

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés 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
- Modelado ágil.
- Gestión de Cambios ágil.
- Refactorización de Base de Datos para mejorar la productividad. Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva. (Maida, Esteban Gabriel & Pacienza, Julián, 2015)

Variación de AUP para la UCI

No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.). En la UCI se diseñó una variación de la metodología AUP, de forma tal que se adaptase al ciclo de vida definido para la actividad productiva en dicha Universidad. (Universidad de las Ciencias Informáticas , 2015)

De las cuatro fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando su objetivo; se unifican las restantes tres fases de AUP en una sola, denominada Ejecución y se agrega la fase de Cierre. (Universidad de las Ciencias Informáticas , 2015)

En la fase de inicio donde se llevan a cabo las actividades relacionadas con la planeación del proyecto. Se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, se realiza

estimaciones de tiempo, esfuerzo y costo y se decide si se ejecuta o no el proyecto. En una la fase de ejecución se producen 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. En la última fase, de cierre, se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Para modelar el negocio en AUP-UCI se proponen las variantes de casos de uso del negocio (CUN), la descripción de proceso de negocio (DPN), modelo conceptual (MC). Después en dependencia del modelo del negocio seleccionado, existen tres formas de especificar los requisitos, casos de uso del sistema (CUS), historias de usuario (HU), descripción de requisitos por proceso (DRP).

Disciplinas Variación AUP-UCI

1. Modelado de negocio
2. Requisitos
3. Análisis y diseño
4. Implementación
5. Pruebas interna
6. Pruebas de liberación
7. Pruebas de Aceptación (Universidad de las Ciencias Informáticas , 2015)

En esta investigación se decide emplear el **escenario 3** que es el que utiliza el centro y así hay una compatibilidad con el sistema, debido a que módulo Trámite Docente pertenece al mismo. Este escenario se aplica al proyecto porque evalúa el negocio a informatizar y como resultado se obtiene un negocio con procesos más complejos. Se tuvo presente que el escenario 3 es muy conveniente a partir de que se desea representar una gran cantidad de niveles de detalles y la relaciones entre los procesos identificados.

1.5 Herramientas y tecnologías a utilizar

Lenguaje de marcado de hipertexto (**HTML 5**) es el lenguaje de marcado predominante en el desarrollo de páginas web, se utiliza en la traducción y descripción de la estructura y la información en forma de texto, así como para complementar este con otros elementos. De manera general, permite definir la estructura y el contenido de las páginas, permitiendo combinar textos, imágenes, sonidos, vídeos y enlaces a otras páginas y entre sus características resalta la reducción de la necesidad de plugins externos y un mejor manejo de errores. (Lapuente, 2019)

TypeScript 4.3

Es un lenguaje transpilado a JavaScript, lo que se traduce y compila antes de ser usado en un navegador. Esto no significa que sea un reemplazo, sino que lo complementa, permitiendo que utilicemos el nuevo código de TypeScript y el lenguaje que ya conocemos de JavaScript. (Negrete, 2022)

El lenguaje TypeScript agrega la escritura estática para estructurar tu código y mejorar las técnicas de programación orientada a objetos. TypeScript tiene la capacidad de detectar errores antes del tiempo de ejecución, por lo que escribirás un código confiable y podrás mitigar las deficiencias de JavaScript. Esta es una de las tantas razones por las que las personas se preguntan qué es TypeScript. (Machuca, 2022)

Hoja de estilo en cascada (CSS 3)

CSS es un lenguaje que sirve para especificar el estilo o aspecto de las páginas web. CSS se define en base a un estándar publicado por una organización llamada W3C, que también se encarga de estandarizar el propio lenguaje HTML. (Desarrollo Web, 2020)

CSS se ideó para aplicar el formato en las páginas, de una manera mucho más detallada, con nuevas posibilidades que no estaban al alcance de HTML. Al mismo tiempo, gracias a la posibilidad de aplicar el estilo de manera externa al propio documento HTML, se consiguió que el mantenimiento de las páginas fuese mucho más sencillo. (Desarrollo Web, 2020)

Python 3.18

Python es un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo. A diferencia de otros lenguajes como Java o .NET, se trata de un lenguaje interpretado, es decir, que no es necesario compilarlo para ejecutar las aplicaciones escritas en Python, sino que se ejecutan directamente por el ordenador utilizando un programa denominado interpretador, por lo que no es necesario “traducirlo” a lenguaje máquina. (Santander Universidades, 2021)

Es uno de los lenguajes de programación más completos y versátiles que existen, ya que no solo es utilizado por desarrolladores y programadores, sino que cada vez tiene más valor para ingenieros y científicos de datos. (Robledano, 2019). En la presente investigación se utiliza para la implementación del backend del sistema.

DJANGO REST FRAMEWORK 3.0

Es un conjunto de herramientas potente y flexible para crear API web. Algunas razones por las que podría querer usar el marco REST: (Django REST framework, n.d.)

- Políticas de autenticación que incluyen paquetes para OAuth1a y OAuth2 .
- Serialización que admite fuentes de datos ORM y no ORM .
- Personalizable hasta el final: solo use las vistas regulares basadas en funciones si no necesita las funciones más potentes .
- Amplia documentación y gran apoyo de la comunidad .
- Utilizado y confiado por empresas reconocidas internacionalmente, como Mozilla, Red Hat , Heroku y Eventbrite.
- La API navegable por la Web es una gran ganancia de usabilidad para sus desarrolladores. Esta fue la razón por la que se utiliza el Django en esta investigación.

Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago. (MDN contributors, 2022)

Angular 8.4

Es un Framework de JavaScript de código abierto escrito en TypeScript. Su objetivo principal es desarrollar aplicaciones de una sola página. Google se encarga del mantenimiento y constantes actualizaciones de mejoras para este framework. (Manuel José Gonçalves , 2021)

Es un framework basado en componentes para crear aplicaciones web escalables. Una colección de bibliotecas bien integradas que cubren una amplia variedad de características, que incluyen enrutamiento, administración de formularios, comunicación cliente-servidor y más. Un conjunto de herramientas para desarrolladores que permiten desarrollar, compilar, probar y actualizar el código fuente de la aplicación. (Manuel José Gonçalves , 2021). Dicho framework es utilizado para el fronted del sistema.

PostgreSQL 14.6

Es el sistema de gestión de base de datos relacional, orientada a objetos de código abierto más avanzado del mundo. Resulta un potente sistema de base de datos objeto/relacional. Cuenta con más de 16 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de confiabilidad, integridad de los datos, y corrección, funciona en todos los principales sistemas operativos, incluyendo

Linux, UNIX y Windows. Es altamente escalable, tanto en la enorme cantidad de datos que puede administrar, como en el número de usuarios concurrentes que puede soportar. (PostgreSQL, 2018). Este gestor de base de datos es quien permite almacenar toda la información relacionada con la gestión de trámites docentes de un estudiante.

Visual Paradigm 17.0

Visual Paradigm es una suite de gestión empresarial y desarrollo de software galardonada líder en todo el mundo, que proporciona todas las funciones que necesita para la arquitectura empresarial, la gestión de proyectos, el desarrollo de software y la colaboración en equipo en una solución integral. (Visual Paradigm, n.d.). Su uso permitió la planificación de la arquitectura empleada, la gestión de proyecto y el desarrollo ágil del sistema, para mantener el control y fomentar el crecimiento.

Git

Es un sistema de control de versiones distribuido gratuito y de código abierto diseñado para manejar todo, desde proyectos pequeños hasta proyectos muy grandes, con rapidez y eficiencia. Git es fácil de aprender y ocupa poco espacio con un rendimiento ultrarrápido. Supera a las herramientas de SCM como Subversion, CVS, Perforce y ClearCase con características como sucursales locales económicas , áreas de preparación convenientes y múltiples flujos de trabajo. (Git, n.d.)

GitLab es una plataforma totalmente integrada basada en Git para el desarrollo de software. Además de las funcionalidades de Git, GitLab tiene un montón de potentes características para mejorar tu flujo de trabajo. (Docs Gitlab, n.d.)

Visual Studio Code (VS Code) 1.43

Es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación. (OpenWebinars, 2022)

1.6 Conclusiones parciales

En este capítulo se aborda sobre los términos asociados al desarrollo de la investigación. Además, de lo referente al proyecto Sistema de Gestión Académico XAUCE AKADEMOS para el MINED, específicamente del módulo “Tramite Docente”

con el cual se trabajará, así como las tecnologías y herramientas a utilizar, proponiendo la metodología (AUP-UCI escenario 3) para su desarrollo. Por otra parte, fue fundamental el estudio de propuestas similares para el desarrollo de la investigación.

Capítulo 2: Propuesta de solución

En este capítulo se explica el proceso de desarrollo de software que conlleva a la solución. Además, se presentan diferentes elementos de la ingeniería de requisitos a utilizar en el sistema propuesto. Se documentan las disciplinas de requisito y de análisis y diseño de la solución propuesta, teniendo en cuenta la metodología de desarrollo software seleccionada. En el caso de la disciplina de requisitos se definen las actividades establecidas para la ingeniería de requerimientos.

2.1 Descripción de la propuesta de solución

En el módulo Trámite Docente 1.1 se realizan los trámites de matriculación, de registro, de baja, de ubicación por continuidad de estudio. Permite al MINED controlar y obtener información sobre los procesos académicos del país y tomar decisiones en un futuro a partir de los resultados de los trámites registrados, por ejemplo, la cantidad de estudiantes, la cantidad de graduados y los traslados.

2.2 Disciplina Modelado de Negocio

El Modelado del Negocio es la disciplina destinada a comprender los procesos de negocio de una organización. Se comprende 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: (Universidad de las Ciencias Informáticas, 2015)

1. Casos de Uso del Negocio (CUN).
2. Descripción de Proceso de Negocio (DPN).
3. Modelo Conceptual (MC).

Un modelo de negocios es un instrumento que permite tener claridad en el modo por el cual se ofrece una solución a una necesidad del mercado. Estos modelos tienen que ser claros y definir las estrategias que deben emplearse en la búsqueda de los objetivos de un negocio. A través de un modelo de negocios se debe distinguir cuál es el valor de un producto, qué se necesita para insertarlo en el mercado y a qué tipo de clientes se venderá. (Camilo Clavijo, 2022)

Modelo conceptual

El modelo conceptual no se ocupa de implementar la base de datos real. Funciona a un nivel muy alto, traduciendo los problemas del mundo real y los requisitos comerciales en un marco conceptual que es fácil de entender y que, por esta razón, puede comunicarse también a los no expertos. Solo se enfoca en identificar las entidades que

serán parte de la base de datos y las relaciones entre ellas, es decir, las operaciones o asociaciones que existen entre ellas. Es muy eficaz para comprender y definir los procesos comerciales de la empresa. (Talent Garden, 2021)

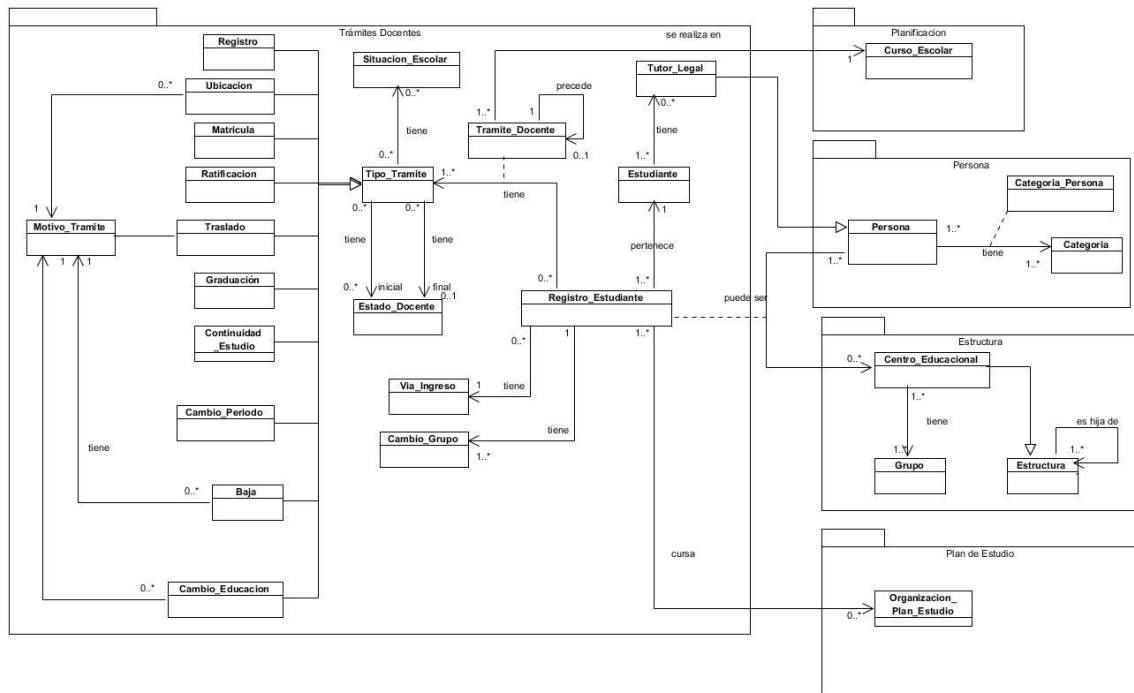


ILUSTRACIÓN 0.1 MODELO CONCEPTUAL

Motivo de trámite: motivo por el cual se le realiza un trámite a un estudiante.

Registro: se registra todos los datos del estudiante.

Ubicación: ubicación escolar del estudiante.

Matricula: listado oficial donde se registran los estudiantes.

Ratificación: el estudiante ratifica sus datos para continuar los estudios.

Traslado: acción que define el traslado de un estudiante a otra entidad.

Graduación: confirmación de graduado en un nivel escolar.

Continuidad de estudio: el estudiante pasa el nivel escolar y continúa estudiando.

Cambio de período: es el cambio de período en un mismo curso escolar.

Baja: cuando el estudiante es baja de una institución escolar.

Cambio de educación: el estudiante cambia de educación con el objetivo de mejorar la misma.

Situación escolar: el estudiante presenta alguna situación durante su paso por el sistema educativo.

Tipo de trámite: diferentes trámites por los que puede pasar un estudiante.

Estado docente: condiciones que presenta un estudiante docentemente.

Vía de ingreso: modo en que ingreso el estudiante a una institución.

Trámite docente: actividades que se realizan para gestionar los trámites de un estudiante.

Estudiante: persona que cursa estudios en un centro docente.

Tutor legal: encargado de un estudiante menor de edad.

Persona: hace referencia a un ser humano.

Curso escolar: período en que un estudiante acude a su centro de enseñanza.

Categoría: posición de una persona o cosa dentro de una clasificación.

Categoría persona: clasificación a la que pertenece una persona.

Centro educacional: unidad básica donde los estudiantes reciben los conocimientos.

Grupo: conjunto de personas con alguna relación en común.

Estructura: conjunto de partes relacionadas a una organización.

Organización plan de estudio: descripción de los elementos que comprenden un plan de estudio y la relación entre ellos.

TABLA 1. REGLAS DEL NEGOCIO. FUENTE: ELABORACIÓN PROPIA

No	Tipo	Nombre	Descripción
1.	Regla textual	Nombre de los grupos de estudio	El nombre de los grupos de estudio debe estar conformados por las siglas de la Especialidad del MINED, tipo de curso, año académico.
2.	Regla textual	Especialidades del MINED de un centro educacional	En un centro educacional del MINED se pueden impartir una o varias especialidades del MINED, esto sin importar de que nivel educativo sean dichas especialidades.

3.	Regla de relación	Grupos de estudio de un centro educacional	Un centro educacional puede tener uno o varios grupos de estudio asociados siempre y cuando sean de las Especialidades del MINED que se imparten en el Centro.
4.	Regla de relación	Claustro de un Centro educacional	Un claustro está definido por un centro educacional o unidad organizativa y un curso escolar.
5.	Regla de relación	Profesores de un claustro	Un claustro puede estar conformado por uno o varios profesores, y estos pueden o no pertenecer (ser trabajadores) al Centro educacional para el que se conforma el claustro.
6.	Regla textual	Asignaturas que imparte el profesor de un claustro	Cuando un profesor forma parte de un claustro deben quedar definidas las asignaturas que va a impartir como parte del mismo.
7.	Regla de relación	Eliminar profesor de un claustro	Cuando un profesor deja de formar parte de un claustro, debe quedar registrado que fue parte del mismo si emitió notas a estudiantes, y debe ser eliminado como profesor de los grupos docentes a los que les impartió clases.
8.	Regla de relación	Grupo docente	Se crea un grupo docente por versión de asignatura (registro del profesor), al cual se le asocian los estudiantes que están matriculados en una organización de plan de estudio que tiene dicha versión de asignatura. Un grupo docente siempre tiene que tener estudiantes matriculados en él.
9.	Regla de relación	Eliminar un grupo docente	Para que un grupo docente deje de existir no puede contener estudiantes matriculados.
10.	Regla textual	Estudiantes de un grupo docente	Un estudiante puede ser matriculado en un grupo docente de una asignatura que ya cursó, esto significa que las notas que se reciban serán remplazadas por

			<p>las nuevas, pero sin eliminar las anteriores.</p> <p>Para que un estudiante pueda formar parte de un grupo docente, debe estar matriculado en el centro educacional al cual pertenece el grupo docente y en su misma Especialidad del MINED (carrera).</p>
11.	Regla de relación	Notas de un estudiante	<p>Ninguna nota recibida por un estudiante en cualquier asignatura puede ser remplazada por otra, al menos que se hay cometido un error en su registro.</p> <p>Solo la nota del tipo final es la que se considera como nota global de una asignatura.</p>
12.	Regla de relación	Conceptos por los que puede relacionarse un estudiante con una evaluación	<p>Un estudiante puede tener una asignatura evaluada o asociada con diferentes conceptos:</p> <ul style="list-style-type: none"> - Abonada (no tiene que dar la asignatura y no tiene nota) - Convalidada (puede ser convalidado con nota o sin nota) - Eximido (tiene nota)
13.	Regla de relación	Eliminar un estudiante de un grupo docente	<p>Un estudiante no puede ser eliminado de un grupo docente si este tiene notas emitidas en la asignatura del grupo, es decir, ya no formará parte de su matrícula, pero debe quedar el registro de que fue su miembro.</p>

Como no han ocurrido cambios en el negocio se reutilizarán los artefactos resultantes de la disciplina modelación de negocio del proyecto donde fue desarrollada la versión 1.0 del módulo Trámites Docentes del Sistema Gestión XAUCE AKADEMOS para el MINED. El estudio de estos artefactos permitió comprender el negocio que gestiona dicho modulo para el posterior desarrollo de la versión 1.1.

2.3 Disciplina 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. (Universidad de las Ciencias Informáticas , 2015)

La Ingeniería de Requisitos es un enfoque sistemático y disciplinado para la educación, documentación, validación y gestión de requisitos de software. Entre sus principales objetivos se encuentran: (Universidad de las Ciencias Informáticas, s.f.)

- Conocer los requisitos relevantes, logrando un consenso entre los "interesados" en estos requisitos,
- Comprender y documentar los deseos y necesidades de los "interesados",
- Especificar y gestionar requisitos para minimizar el riesgo de entregar un sistema que no cumpla con los deseos y necesidades de los "interesados".

Requisitos funcionales

Los requerimientos funcionales de un sistema, son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones. Por lo general, estos deben incluir funciones desempeñadas por pantallas específicas, descripciones de los flujos de trabajo a ser desempeñados por el sistema y otros requerimientos de negocio, cumplimiento, seguridad u otra índole. (PMOinformatica, 2017). [Visualizar descripciones de los requisitos funcionales en documento adjunto.](#)

TABLA 2 REQUISITOS FUNCIONALES. ELABORACIÓN PROPIA

Nº	Nombre	Descripción
RF 1.	Ver detalles del estudiante	El sistema debe mostrar los datos del estudiante seleccionado.
RF 2.	Listar estudiantes	El sistema debe mostrar un listado de estudiantes organizado alfabéticamente.
RF 3.	Ver listado de trámites docentes de un estudiante	El sistema debe permitir ver el listado de tramites docente del estudiante seleccionado.
RF 4.	Registrar trámite de registro	El sistema debe permitir registrar el trámite de registro.
RF 5.	Modificar trámite de registro	El sistema debe permitir modificar el trámite de registro seleccionado.
RF 6.	Ver datos de un trámite de registro	El sistema debe mostrar los detalles del trámite de registro seleccionado.

RF 7.	Ver datos de un trámite de ubicación	El sistema a debe mostrar los detalles del trámite de ubicación.
RF 8.	Registrar trámite de ubicación por registro	El sistema debe permitir registrar el trámite de ubicación por registro.
RF 9.	Modificar trámite de ubicación por registro	El sistema debe permitir modificar el trámite de ubicación por registro.
RF 10.	Registrar trámite de ubicación por traslado	El sistema debe permitir registrar el trámite de ubicación por traslado.
RF 11.	Modificar trámite de ubicación por traslado	El sistema debe permitir modificar el trámite de ubicación por traslado.
RF 12.	Registrar trámite de ubicación por cambio	El sistema debe permitir registrar el trámite de ubicación por cambio.
RF 13.	Modificar trámite de ubicación por cambio	El sistema debe permitir modificar el trámite de ubicación por cambio.
RF 14.	Registrar trámite de ubicación por continuidad de estudio	El sistema debe permitir registrar el trámite de ubicación por continuidad de estudio.
RF 15.	Modificar trámite de ubicación por continuidad de estudio	El sistema debe permitir modificar el trámite de ubicación por continuidad de estudio.
RF 16.	Registrar trámite de ubicación por baja	El sistema debe permitir registrar el trámite de ubicación por baja.
RF 17.	Modificar trámite de ubicación por baja	El sistema debe permitir modificar el trámite de ubicación por baja.

RF 18.	Registrar trámite de matrícula	El sistema debe permitir registrar el trámite de matrícula.
RF 19.	Modificar trámite de matrícula	El sistema debe permitir modificar el trámite de ubicación por matrícula.
RF 20.	Ver datos de un trámite de matrícula.	El sistema debe mostrar los detalles del trámite matrícula seleccionado.
RF 21.	Registrar trámite de traslado	El sistema debe permitir registrar el trámite de traslado.
RF 22.	Modificar trámite de traslado	El sistema debe permitir modificar el trámite de traslado por matrícula.
RF 23.	Ver datos de un trámite de traslado	El sistema debe mostrar los detalles del trámite traslado seleccionado.
RF 24.	Registrar trámite de cambio de educación	El sistema debe permitir registrar el trámite de cambio de educación.
RF 25.	Modificar trámite de cambio de educación	El sistema debe permitir modificar el trámite de cambio de educación.
RF 26.	Ver datos de un trámite de cambio de educación	El sistema debe mostrar los detalles del trámite de cambio de educación seleccionado.
RF 27.	Registrar trámite de baja	El sistema debe permitir registrar el trámite de baja.
RF 28.	Modificar trámite de baja	El sistema debe permitir modificar el trámite de baja.
RF 29.	Ver datos de un trámite de baja	El sistema debe mostrar los detalles del trámite de baja seleccionado.

Requisitos no funcionales

Requisitos de eficiencia de desempeño

RNF 1: El sistema debe responder las peticiones del usuario en un tiempo inferior o igual a 3 segundos.

Requisitos de Usabilidad

RNF 1: El sistema debe cumplir con las pautas de diseño establecidas en la Estrategia Marcaria de la UCI.

RNF 2: El sistema debe poseer un diseño “Responsive”.

RNF 3: El sistema debe permitir que se accedan a las funcionalidades con 3 clics como máximo.

RNF 4: El sistema debe pedir la confirmación al usuario antes de realizar una acción crítica

RNF 5: El sistema debe poseer iconos que sugieran las acciones a realizar.

RNF 6: Las vistas del sistema deben ser homogéneas en cuanto a la distribución, colores y diseño de gráficos.

RNF 7: El sistema debe proveer retroalimentación o respuesta a cada una de las acciones realizadas por el usuario.

RNF 8: El sistema debe mostrar mensajes de error indicando la causa del mismo.

Requisitos de Seguridad

RNF 1: El Sistema debe garantizar que los usuarios accedan solo a aquellas acciones que pueden realizar según los roles y permisos que posean.

RNF 2: El Sistema debe validar los datos antes de almacenarlos.

RNF 3: El Sistema debe prevenir el acceso no autorizado a través de URL.

RNF 4: El Sistema debe registrar la trazabilidad de los datos y eventos.

Requisitos de portabilidad

RNF 1: El sistema debe funcionar de forma exitosa en un entorno de hardware y software.

2.4 Disciplina Análisis y diseño

En esta disciplina, si se considera necesario, 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 (Universidad de las Ciencias Informáticas , 2015).

2.4.1 Arquitectura de la solución

La arquitectura del software por tanto define la estructura que debe de tener un software, las piezas que debemos construir y el modo en el que se deben de juntar y trabajar entre ellas. Se define a alto nivel mediante una serie de patrones y abstracciones que seguir para el desarrollo del software y para la interacción entre sus diversas piezas. (Fernán García de Zúñiga, 2020)

Para la implementación de la solución se trabajó con la arquitectura que ya presentaba el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED, arquitectura de microservicios, la cual surge debido a que toda la información está alojada en un servidor, por lo que no hay separación entre módulos y las diferentes partes del programa están muy acopladas. De este modo se soluciona la necesidad por parte de las empresas de realizar cambios en el software e implementarlos de forma fácil y rápida.

Los microservicios han permitido crear infraestructuras (IT) más adaptables y flexibles, de manera que, si se desea modificar solamente un servicio, no es necesario alterar el resto de la infraestructura. Cada uno de los servicios se puede desplegar y modificar sin que ello afecte a otros servicios o aspectos funcionales de la aplicación, facilitando el trabajo del desarrollador.

Docker es una tecnología que facilita la implementación de arquitecturas de microservicios, donde los encargados del proyecto puedan desarrollar, desplegar y ejecutar aplicaciones en un entorno aislado denominado contenedor. Docker permite separar las aplicaciones de la infraestructura acelerando el proceso de entrega de software a producción. Debido a ello, el sistema es desarrollado y desplegado teniendo en cuenta las bondades que brinda la plataforma Docker.

Patrón arquitectónico

Multicapas: se caracteriza por organizar los componentes del sistema en capas con responsabilidades bien definidas y donde las capas del nivel más alto invocan los servicios de las del nivel inferior. En el sistema es posible identificar las siguientes capas: infraestructura, presentación, negocio, acceso a datos y datos.

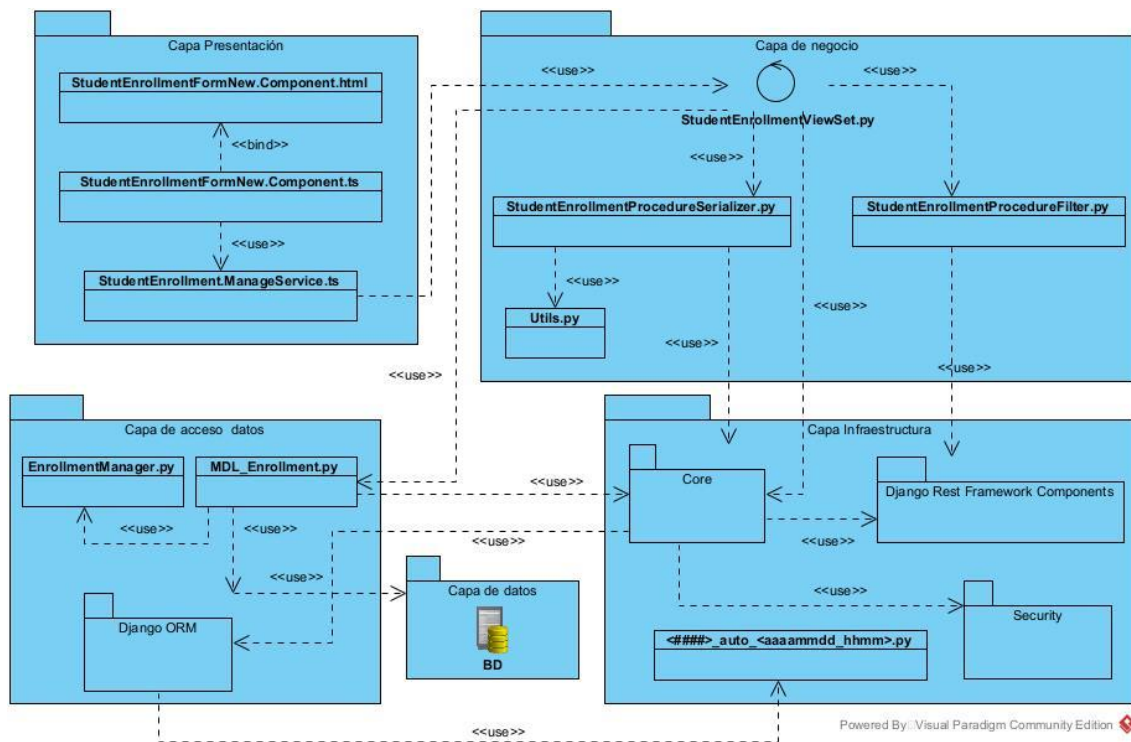


ILUSTRACIÓN 0.2 DIAGRAMA DE ANÁLISIS DEL DISEÑO. REGISTRAR TRÁMITE DE MATRÍCULA

Capa de presentación: esta capa es la encargada de representar la información deseada y la entrada de datos.

Capa de negocio: esta capa se encarga de procesar la información a insertar o mostrar y provee los servicios web. Maneja la información proveniente tanto de la capa de acceso a datos como de la capa de presentación.

Capa de acceso a datos: proporciona acceso simplificado a los datos almacenados en la base de datos del sistema.

Capa de infraestructura: esta capa proporciona las migraciones, los componentes del framework, la seguridad y el Core de la aplicación.

Capa de datos: en esta capa se encuentra la base de datos del sistema.

Patrones de Diseño

Los patrones de diseño, son una solución general, reutilizable y aplicable a diferentes problemas de diseño de software. Se trata de plantillas que identifican problemas en el sistema y proporcionan soluciones apropiadas a problemas generales a los que se han enfrentado los desarrolladores durante un largo período de tiempo, a través de prueba y error (Miriam Martínez Canelo , 2020). Los patrones de diseño pueden ser de dos tipos: GRASP (Patrones Generales de Software para Asignación de Responsabilidades)

y GoF (Pandilla de Cuatro). A continuación, se presentan los patrones utilizados en el sistema.

Patrones GRAPS

Este patrón brinda un enfoque sistemático, racional y explicable para aprender el diseño de objetos. Este enfoque se basa en los patrones de asignación de responsabilidades.

Experto: Asignar una responsabilidad al experto en información. Este patrón se refleja en la clase *Enrollment_Procedure*.

```
152
153     # update person data
154     person_obj = Person.objects.get(pk=person_id)
155
156     update_person_data(person_obj, height=data.get('height'), weight=data.get('weight'))
157     update_procedure_student(procedure_student_obj, data.get('orphan_id'), data.get('dis
158
159     # save diseases of students
160     # se accede a los metodos de la clase procedurereasons a través del objeto ya creado
161     if diseases:
162         procedure_student_obj.diseases.set([])
163         procedure_student_obj.save()
164         associate_students_and_diseases(procedure_student_obj, diseases)
165
166     # pdf
167     #
168     # try:
169     #     async_control_enrollment_pdf(person_obj, student_registration_obj, student_pro
170     #     procedure_enrollment_obj)
171     # except Exception:
172     #     pass
173
174     # transaction.savepoint_commit(sid)
175
176     # serializer data for student registration
177     student_registration = StudentRegistrationSerializer(student_registration_obj)
178
179     return student_registration.data
```

ILUSTRACIÓN 0.3 FRAGMENTO DE CÓDIGO REFERENTE AL PATRÓN EXPERTO

Bajo Acoplamiento: Un elemento con bajo acoplamiento no depende de muchos elementos, uno con alto acoplamiento si lo hace, lo cual puede llevar problemas como que los cambios en las clases relacionadas fueren cambios locales, o que las clases sean difíciles de usar y de entender de manera aislada. Se ve reflejado en la clase *Procedure_Reason*.

```
server > procedure > api > views > view_procedure_reason.py
1 from core.api.generics.generic import BaseAPIViewSet
2 from django_filters import rest_framework as dj_filters
3 from procedure.api.filters import ProcedureReasonFilter
4 from procedure.api.serializers import ProcedureReasonSerializer
5 from procedure.models import ProcedureReason
6 from rest_framework import filters
7 from rest_framework_datatables.filters import DatatablesFilterBackend
8
9
10 class ProcedureReasonViewSet(BaseAPIViewSet):
11     queryset = ProcedureReason.objects.all().order_by('is_disable', 'name')
12     serializer_class = ProcedureReasonSerializer
13     filter_backends = (dj_filters.DjangoFilterBackend, filters.SearchFilter, filters.Orderin
14                       DatatablesFilterBackend)
15     filter_class = ProcedureReasonFilter
16
```

ILUSTRACIÓN 0.4 FRAGMENTO DE CÓDIGO REFERENTE AL PATRÓN BAJO ACOPLAMIENTO

Alta Cohesión: Incrementa la claridad y facilita la comprensión. Implica el mantenimiento y las mejoras, soporta a menudo bajo acoplamiento. Se muestra en la clase *Student_Enrollment*.

```

"model": "procedure.studentproceduretype",
"pk": 1,
"fields": {
  "name": "Registro",
  "is_disable": false,
  "create_date": "2021-03-11T20:19:42.662Z",
  "update_date": "2021-03-11T20:19:42.662Z",
  "description": "Configuracion de registro",
  "changes_state": false,
  "changes_scholar_situation": false,
  "changes_structure": false,
  "final_state": 1,
  "procedure_type": "REGISTRATION",
  "initial_states": [],
  "school_situations": []
}

```

ILUSTRACIÓN 0.5 FRAGMENTO DE CÓDIGO REFERENTE AL PATRÓN ALTA COHESIÓN

Patrones GOF

El objetivo principal de los patrones es facilitar la reutilización de diseños y arquitecturas software que han tenido éxito capturando la experiencia y haciéndola accesible a los no expertos.

Patrón de creación: Trata de la inicialización y configuración de clases y objetos. Se utiliza desde la clase *Way_Entry_Serializer*.

```

server > procedure > api > serializers > srl_way_entry.py
1 from core.api.generics.generic import BaseAkModelSerializer
2 from procedure.models import WayEntry
3 from rest_framework import serializers
4
5
6 class WayEntrySerializer(BaseAkModelSerializer):
7     name = serializers.CharField(required=True)
8
9     class Meta:
10        model = WayEntry
11        fields = ('id', 'name', 'description', 'is_disable')
12        read_only_fields = ('id', 'type')
13

```

ILUSTRACIÓN 0.6 FRAGMENTO DE CÓDIGO REFERENTE AL PATRÓN CREADOR

Patrón estructural: Trata de desacoplar interfaz e implementación de clases y objetos. Se utiliza desde el componente *Way_Entry_Form*.

```

1 import { Component, OnInit } from '@angular/core';
2 import { FormBuilder, FormGroup, Validators } from '@angular/forms';
3 import { ActivatedRoute, Router } from '@angular/router';
4 import { Ng4LoadingSpinnerService } from 'ng4-loading-spinner';
5 import { Observable } from 'rxjs';
6 import { AkMessagesServiceService } from 'src/app/core/utils/akMessagesService.service';
7 import { ApplicatioMessages } from 'src/app/core/utils/messages/applicationMessages';
8 import { AkCustomValidator } from 'src/app/core/_helpers/validators/ak-custom-validator';
9 import { FormErrorsService } from 'src/app/core/_services/form-error.service';
10 import { WayEntryManageService } from 'src/app/modules/procedure/manager/way-entry.service';
11
12 @Component({
13   selector: 'app-way-entry-form',
14   templateUrl: './way-entry-form.component.html',
15   styleUrls: ['./way-entry-form.component.scss'],
16 })
17 export class WayEntryFormComponent implements OnInit {
18   data: any = {};
19   createForm: FormGroup;
20   applicationMessages = ApplicatioMessages;
21   title = '';
22   errorsFromApi$: Observable<any[]>;
23
24   constructor(
25     private fb: FormBuilder,
26     private router: Router,
27     private route: ActivatedRoute,
28     public manager: WayEntryManageService,
29     private message: AkMessagesServiceService,
30     private loadingSpinnerService: Ng4LoadingSpinnerService,

```

ILUSTRACIÓN 0.7 FRAGMENTO DE CÓDIGO REFERENTE AL PATRÓN ESTRUCTURAL

Patrón de comportamiento: Trata de las interacciones dinámicas entre sociedades de clases y objetos. Se utiliza en la clase *Worker_Procedure*.

```
111         return worker_procedure_register_obj
112
113     # update worker registration
114     worker_registration_obj = update_worker_registration(
115         worker_registration_obj=worker_registration_obj,
116         previous_procedure_id=worker_procedure_obj.id)
117
118     if isinstance(worker_registration_obj, JsonResponse):
119         return worker_registration_obj
120
121
122     # update available positions
123     try:
124         structure_responsibility_obj = StructureResponsibility.objects.filter(
125             responsibility_id=worker_registration_obj.responsibility.id,
126             structure_id=worker_registration_obj.structure.id).first()
127         structure_responsibility_obj.ocupada += 1
128         structure_responsibility_obj.save()
129
130     except ObjectDoesNotExist as err:
131         return JsonResponse({'non_field_errors': str(err)}, status=status.HTTP_400_BAD_REQUEST)
132
133
134     # save person structure relationship
135     save_person_structure(category_id, worker_registration_obj)
136
137     # serializer data for worker registration
138     worker_registration = WorkerRegistrationSerializer(
139         worker_registration_obj)
```

ILUSTRACIÓN 0.8: FRAGMENTO DE CÓDIGO REFERENTE AL PATRÓN DE COMPORTAMIENTO

2.5 Conclusiones parciales

Con el uso de la metodología AUP-UCI en su escenario 3, se pudo organizar de manera correcta el desarrollo de la solución y generar los productos de trabajos necesarios, tales como la arquitectura empleada y el patrón arquitectónico utilizado, todo esto correspondiente a la disciplina de análisis y diseño a través de la descripción de la arquitectura del sistema. Además, del levantamiento de requisitos realizado, los cuales darán respuesta a las necesidades plasmadas. Por otra parte, los patrones de diseño permitieron obtener una solución flexible a la hora de introducir cambio.

Capítulo 3: Implementación y pruebas

El presente capítulo refleja la implementación del módulo desarrollado. Muestra la estrategia de pruebas de software aplicada, la cual refleja los niveles de pruebas aplicados, los tipos de pruebas, métodos de pruebas y técnicas de pruebas que se utilizaron. Se muestran los resultados de las pruebas realizadas dejando ver el correcto funcionamiento del módulo actualizado.

3.1 Disciplina implementación

En la implementación, a partir de los resultados del Análisis y Diseño se construye el sistema. En esta disciplina el nivel de prueba que se ejecuta es el de componente y se realizan pruebas unitarias.

3.3.1 Diagrama de despliegue

Un diagrama de despliegue es un tipo de diagrama UML que muestra la arquitectura de ejecución de un sistema, incluyendo nodos como entornos de ejecución de hardware o software. Los diagramas de despliegue se utilizan para visualizar el hardware y el software físico de un sistema. Con su uso se puede entender cómo el sistema se desplegará físicamente en el hardware. (Creately, 2022)

El diagrama de despliegue permite modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes (nodos). Los nodos son objetos físicos que existen en tiempo de ejecución y que representan algún tipo de recurso computacional, también pueden ser dispositivos del sistema.

El diagrama de despliegue de la solución propuesta se representa por una PC cliente que solicita peticiones mediante el protocolo HTTP al servidor de aplicaciones Web, el cual se conecta al servidor de base de datos PostgreSQL para extraer la información, como se muestra en siguiente figura.

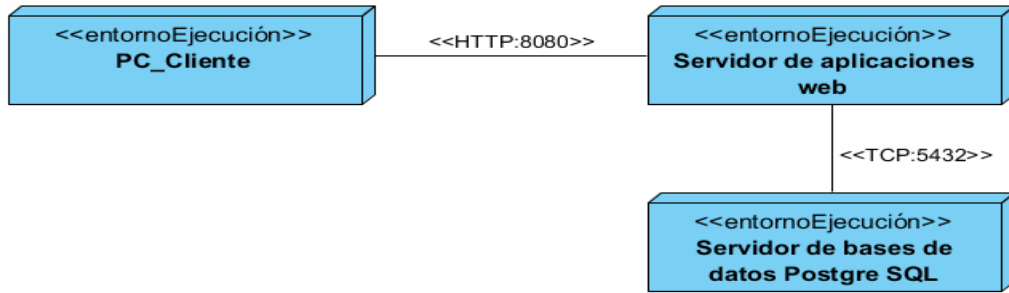


ILUSTRACIÓN 0.1 DIAGRAMA DE DESPLIEGUE

3.1.2 Diagrama de componente

Los diagramas de componentes UML representan las relaciones entre los componentes individuales del sistema mediante una vista de diseño estática. Pueden ilustrar aspectos de modelado lógico y físico. Un diagrama de componentes proporciona una visión general del sistema y documenta la organización de los componentes del sistema y sus relaciones y dependencias mutuas. Los diagramas de componentes proporcionan una visión orientada a la ejecución, es decir, dan al desarrollador información sobre si un sistema funciona de forma coherente y cumple sus tareas y objetivos. (ionos, 2020)

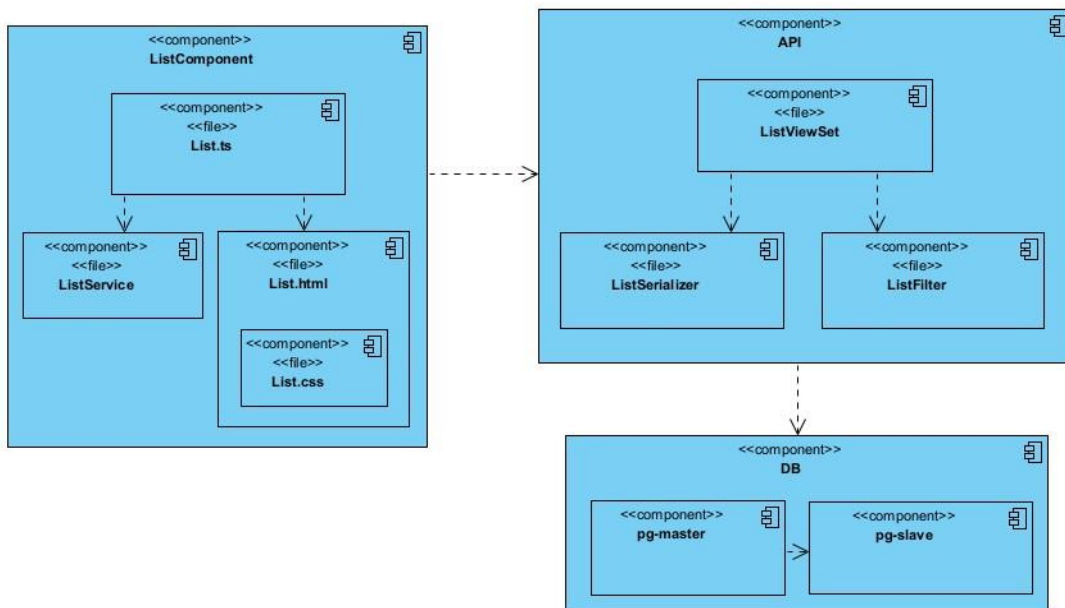


ILUSTRACIÓN 0.2 DIAGRAMA DE COMPONENTE

3.2 Estrategia de prueba

Una estrategia de prueba de software proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos se requerirán. Por tanto, cualquier

estrategia de prueba debe incorporar la planificación, el diseño de casos de prueba, la ejecución de la prueba, la recolección y evaluación de los resultados. Una estrategia de prueba de software debe ser suficientemente flexible para promover un uso personalizado de la misma. Igualmente, debe ser suficientemente rígida para alentar la planificación razonable y el seguimiento de la gestión conforme avanza el proyecto. (Pressman R. , 2010)

De igual manera, una estrategia de pruebas de software debe establecer pruebas de bajo nivel para verificar que un pequeño segmento de código fuente se haya implementado correctamente. Además, debe establecer pruebas de alto nivel que validan las principales funciones del sistema contra requerimientos del cliente. Estas integran los niveles, tipos, métodos y técnicas de diseño de casos de prueba en una serie de pasos bien planificados que llevan a la construcción correcta del software. Constituyen una parte fundamental del proceso de validación y verificación del software. La verificación es una actividad que permite asegurar que las distintas partes del software cumplen con la función para la cual fueron diseñadas. En este sentido la verificación se encarga de revisar el funcionamiento de los módulos del software, mientras que la validación se encarga de comprobar que los módulos verificados cumplen con los requisitos que el cliente ha expresado. (Pressman & Maxim, 2015)

A continuación, se refleja la estrategia de prueba utilizada para corroborar que la solución implementada es válida y que cumple con los requisitos acordados.

Tabla 3. Estrategia De Prueba: Elaboración Propia

Niveles de Prueba	Tipo de Pruebas	Método	Técnica
Componentes	Unitarias	Caja Blanca	Camino básico
Integración	Funcional	Caja Negra	Partición de equivalencia
	Usabilidad		Evaluación heurística(lista de chequeo)
	Seguridad		Evaluación heurística(lista de chequeo)
	Rendimiento		
Aceptación	Alfa		Partición de equivalencia

3.3 Pruebas de Software

Las pruebas de software son un conjunto de procesos con los que se pretende probar un sistema o aplicación en diferentes momentos para comprobar su correcto funcionamiento. Este tipo de pruebas abarca cualquier estadio del desarrollo del sistema, desde su creación hasta su puesta en producción. Lo interesante de las pruebas es que se puedan ejecutar de manera automática, para determinar en cualquier momento si tenemos una aplicación estable o si, por el contrario, un cambio en una parte ha afectado a otras partes sin que nos demos cuenta. (Jorge Turrado, 2020).

3.3.1 Pruebas unitarias

Estas pruebas generalmente son realizadas por el desarrollador y automatizadas. Se enfocan en los componentes, unidades o módulos, es decir, los elementos más pequeños del software. Algunos objetivos de estas pruebas son verificar que los comportamientos funcionales y no funcionales del componente son los diseñados y especificados, encontrar defectos, prevenir la propagación de defectos en otros niveles de prueba. (Carla Gomez, 2020)

Para la realización de estas pruebas se utiliza el **método de caja blanca**, la cual pretende investigar sobre la estructura interna del código, exceptuando detalles referidos a datos de entrada o salida, para probar la lógica del programa desde el punto de vista algorítmico. Se realiza un seguimiento del código fuente según se va ejecutando los casos de prueba, determinándose de manera concreta las instrucciones, bloques, etc. que han sido ejecutados por los casos de prueba. En las pruebas de Caja Blanca se desarrollan casos de prueba que produzcan la ejecución de cada posible ruta del programa o módulo, considerándose una ruta como una combinación específica de condiciones manejadas por un programa. (Eduardo Salazar Martínez, 2015).

Se utiliza la **técnica de camino básico**, la cual nos permite obtener una medida de la complejidad lógica de un diseño, y usar esta medida como guía para la definición de un conjunto básico. Además, de que la idea es, derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control.

Con el objetivo de valorar la calidad de la implementación fue necesario aplicar la técnica descrita anteriormente: técnica de camino básico. Para ello se siguieron los siguientes pasos:

- A partir del código fuente, dibujar el grafo de flujo asociado.

- Calcular la complejidad ciclomática del grafo.
- Determinar el conjunto básico de caminos independientes.
- Preparar los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Dando cumplimiento a los pasos básicos, se enumeran cada una de las sentencias de código de la funcionalidad validar formulario, que basicamente valida si se puede o no guardar los datos de un formulario

```

1 validSave() {
2   let valid = true;

   Object.keys(this.mapOfCheckedId).forEach((key) => {

3     4 if (this.mapOfCheckedId[key]) {
5       const foundInActualData = this.listOfAllData.find(e => e.registration_id === key);
6       if (foundInActualData) {
7         if (foundInActualData.school_situation_id === null || foundInActualData.academic_year_id === null || foundInActualData.schoo
8           valid = false;
9           return valid;
10        }
11       } else {
12        const foundInAllData = this.allDataVerify.find(e => e.registration_id === key);
13        if (foundInAllData.school_situation_id === null || foundInAllData.academic_year_id === null || foundInAllData.school_period_
14          valid = false;
15          return valid;
16        }
17      }
18    });
19  }
20  return valid;
21 }

```

ILUSTRACIÓN 0.3 FRAGMENTO DEL CÓDIGO VALIDAR FORMULARIO

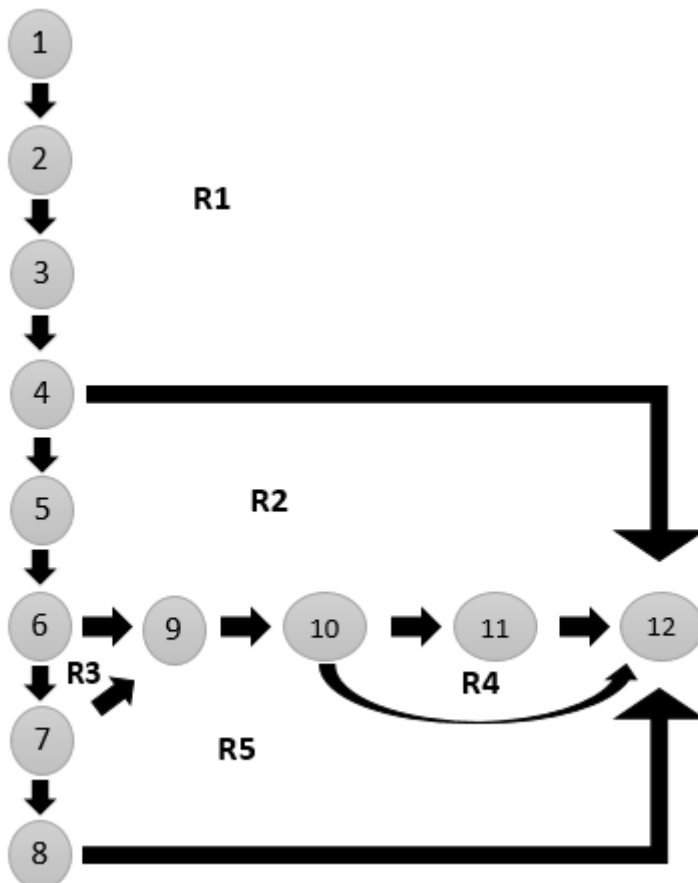


Ilustración 0.4 Grafo De Flujo Asociado Al Método Validar Formulario

La Complejidad Ciclomática, es una métrica del software ampliamente difundida y aceptada, que permite, obtener una medición cuantitativa de la complejidad lógica de un programa o pieza de software. Tiene la particularidad de haber sido desarrollada de manera que pueda aplicarse con independencia del lenguaje en que se ha programado. (Huihui Liu; Xufang Gong; Li Liao; Bixin Li, 2018)

Esta métrica se basa en el diagrama de flujo, determinado por las estructuras de control del código correspondiente a un programa. Del análisis Refactorización de Código y Consideraciones sobre la Complejidad Ciclomática, se puede obtener una medida cuantitativa de la cantidad mínimas de pruebas que se deberían realizar para recorrer al menos una vez cada camino básico del diagrama de flujo. Si se logra establecer un lote de pruebas que realice esto, habremos logrado ejecutar y probar cada sentencia del programa. Se asume que a mayor cantidad de caminos básicos mayor será la complejidad lógica del programa. (Escobar Lascarro, Jesús David, 2019) (García Pérez, Carlos Rubén, 2018)

Si bien el nombre de “Complejidad Ciclomática” es con el que se la conoce, la métrica no se concentra en contar ciclos o bucles sino en las decisiones (condiciones), por lo cual, en realidad lo que estaríamos midiendo, es la cantidad de decisiones simples asumiendo que estas están directamente ligadas a la complejidad lógica del código.

Se calcula de la siguiente manera:

1. $V(G) = E - N + 2$ donde E es el número de aristas y N los vértices.

$$V(G) = 15 - 12 + 2$$

$$V(G) = 5$$

2. $V(G) = P + 1$ siendo “P” la cantidad de nodos predicados (son los nodos de los cuales parten dos o más aristas).

$$V(G) = 4 + 1$$

$$V(G) = 5$$

3. $V(G) = R$ siendo “R” la cantidad total de regiones, se incluye el área exterior del grafo, contando como una región más

$$V(G) = 5$$

El cálculo efectuado mediante las fórmulas antes presentadas muestra una complejidad ciclomática de valor cinco, de manera que existen cinco posibles caminos por donde el flujo puede circular, este valor representa el número mínimo de casos de pruebas para el procedimiento tratado.

Camino básico 1: 1-2-3-4-5-6-7-9-10-11-12

Camino básico 2: 1-2-3-4-5-6-9-10-11-12

Camino básico 3: 1-2-3-4-5-6-7-8-12

Camino básico 4: 1-2-3-4-12

Camino básico 5: 1-2-3-4-5-6-9-10-12

Luego se definen los casos de prueba para comprobar la ejecución de cada camino del conjunto básico. En el diseño de los casos de prueba se deben especificar los siguientes elementos:

Descripción: contiene una descripción sobre las restricciones de los datos de entrada que debe tener el caso de prueba.

Condición de ejecución: se especifican los parámetros que debe poseer el caso de prueba para que se cumpla una condición deseada como respuesta del funcionamiento del procedimiento.

Resultados esperados: se explica el resultado esperado de la ejecución del procedimiento.

La tabla muestra el diseño de caso de prueba para el camino 1 del conjunto básico de caminos linealmente independientes, correspondiente a la funcionalidad validar formulario.

TABLA 4. DISEÑO DE CASO DE PRUEBA PARA EL CAMINO 1: ELABORACIÓN PROPIA

Descripción	Los datos de entrada son válidos y deben cumplir con el formato especificado
Condición de ejecución	Datos válidos: Que los datos entrados sean válidos.
Entrada	Lista de datos

Resultados esperados

Se verifica si se puede guardar los datos de un formulario.

Aplicándose el caso de prueba expuesto anteriormente se evidencia que el flujo de trabajo de la funcionalidad es correcto, debido a que se comprueba que cada sentencia es ejecutada al menos una vez, cumpliéndose las condiciones de la prueba y el resultado esperado es satisfactorio.

3.4 Disciplina Pruebas Internas

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. 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. (Universidad de las Ciencias Informáticas , 2015)

En esta disciplina se ejecuta el nivel de **pruebas de integración**. Las mismas implican probar diferentes módulos de una aplicación de software como grupo. Una aplicación de software se compone de diferentes sub módulos que trabajan juntos para diferentes funcionalidades. El propósito de las pruebas de integración es validar la integración de diferentes módulos juntos e identificar los errores y problemas relacionados con ellos. (Consejos técnicos, 2020). Dentro de este nivel de pruebas se encuentran varios tipos de pruebas, entre ellos:

Las pruebas de rendimiento: son aquellas pruebas que someten a un sistema a una carga de trabajo con el fin de medir su velocidad, fiabilidad y estabilidad en esas condiciones de trabajo. (Herrera, 2019). Se utiliza la herramienta JMeter 18.04 para Ubuntu. El ambiente estuvo conformado por:

- Sistema Operativo: Linux 18
- Microprocesador: Intel® Celeron CPU 3215U @ 1.70GHz
- Memoria RAM: 4GB
- Disco Duro: 500GB

Los resultados de las pruebas de rendimiento se consideran satisfactorios, debido a que los tiempos de respuesta del servidor a cada usuario, ante la interacción de 200 y 500 usuarios concurrentes utilizando las principales funciones del módulo, se encontraron

en el rango de tiempo de 1 a 3 segundos. Con ello queda demostrado que la propuesta de solución es estable, ya que presta servicios todo el tiempo, sin incurrir en fallos.

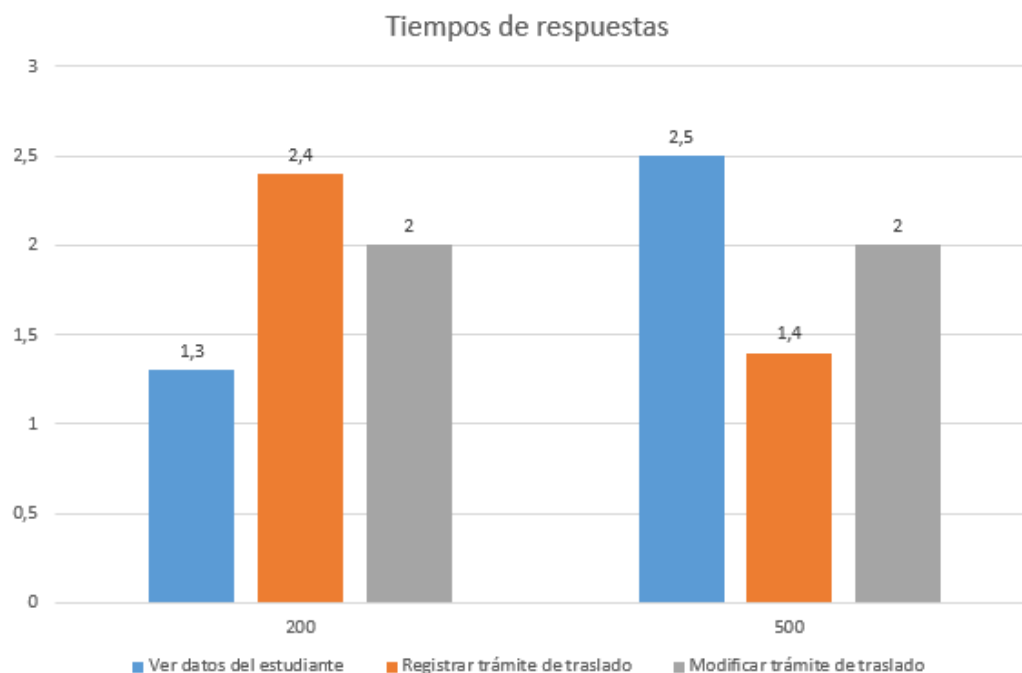


ILUSTRACIÓN 0.5 RESULTADOS DE LA PRUEBAS DE RENDIMIENTO

Las pruebas funcionales: se llevan a cabo para comprobar las características críticas para el negocio, la funcionalidad y la usabilidad. Las pruebas funcionales garantizan que las características y funcionalidades del software se comportan según lo esperado sin ningún problema. Valida principalmente toda la aplicación con respecto a las especificaciones mencionadas en el documento Software Requirement Specification (SRS). Los tipos de pruebas funcionales incluyen pruebas unitarias, pruebas de interfaz, pruebas de regresión, además de muchas. (Consejos técnicos, 2020).

Se utiliza el **método de caja negra**, el cual pretenden examinar el programa en busca de que cuente con las funcionalidades que debe tener y como lleva a cabo las mismas, analizando siempre los resultados que devuelve y probando todas las entradas en sus valores válidos e inválidos. Se desarrollan casos de prueba reales para cada condición o combinación de condiciones y se analizan los resultados que arroja el sistema para cada uno de los casos. (Eduardo Salazar Martínez, 2015)

Para la aplicación de esta prueba se selecciona la **técnica de Partición de Equivalencia**, la cual nos permite examinar los valores válidos e inválidos de las entradas existentes en el software.

Diseño de casos de pruebas

Un caso de prueba es exactamente lo que parece, un escenario de prueba que mide la funcionalidad en un conjunto de acciones o condiciones para verificar el resultado esperado. Se aplican a cualquier aplicación de software, pueden utilizar pruebas manuales o una prueba automatizada y puede hacer uso de herramientas de gestión de casos de prueba. (William Mc Mullin, 2021)

Caso de prueba: Registrar trámite de ubicación por registro.

Como parte de todo el proceso de pruebas, se realizan tres iteraciones de pruebas hasta no detectarse nuevas no conformidades. En la figura se muestra el resultado de las no conformidades detectadas en cada una de las iteraciones de casos de pruebas. [Ver casos de pruebas en documento adjunto.](#)

Durante la aplicación de las pruebas se detectan una serie de no conformidades, las cuales en su mayoría fueron asociadas a errores en las funcionalidades y otras asociadas a errores ortográficos. En una primera iteración se identifican 10 no conformidades, que fueron solucionadas, en una segunda iteración se detectaron 4 no conformidades diferentes a las de la primera iteración realizada, las cuales fueron corregidas y posteriormente en una tercera iteración no se identifican no conformidades.

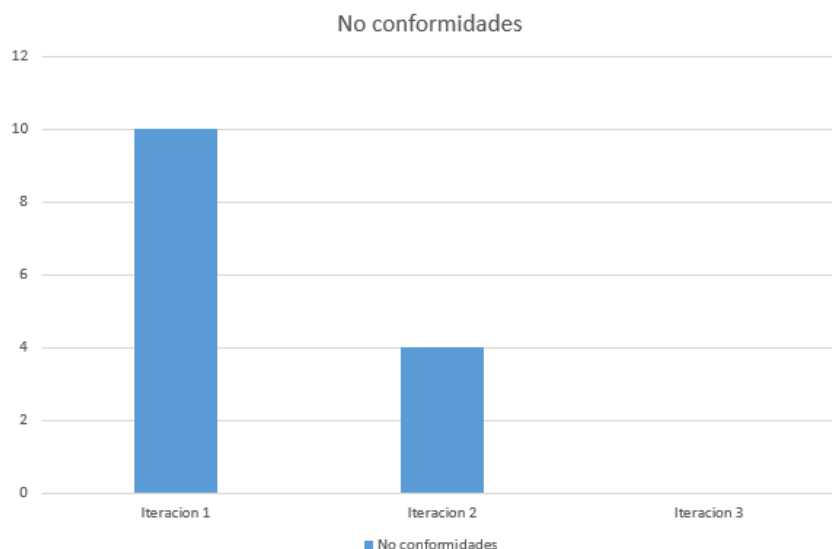


ILUSTRACIÓN 0.6 NO CONFORMIDADES IDENTIFICADAS POR ITERACIONES DE PRUEBA

Las pruebas de usabilidad son un método para evaluar la experiencia del usuario de un producto o sitio web. Generalmente se llevan a cabo estudios de usabilidad desde el

desarrollo inicial del producto hasta su lanzamiento. Esto permite descubrir cualquier problema con el producto a partir de la experiencia del usuario, decidir cómo solucionar estos problemas y, en última instancia, determinar si el producto es realmente útil. (Shelley Pursell, 2021). Se toma como técnica la evaluación heurística para estas pruebas.

Durante la aplicación de las pruebas de usabilidad se detectan una serie de no conformidades, las cuales en su mayoría fueron asociadas a la accesibilidad por parte del usuario al sistema y otras con la estética de la interfaz de usuario. En una primera iteración se identifican 12 no conformidades, que fueron solucionadas, en una segunda iteración se detectan 7 no conformidades distintas a de la primera iteración, las cuales fueron corregidas y posteriormente en una tercera iteración no se identifican no conformidades. [Anexo 1 : Lista de chequeo para la prueba de usabilidad](#)

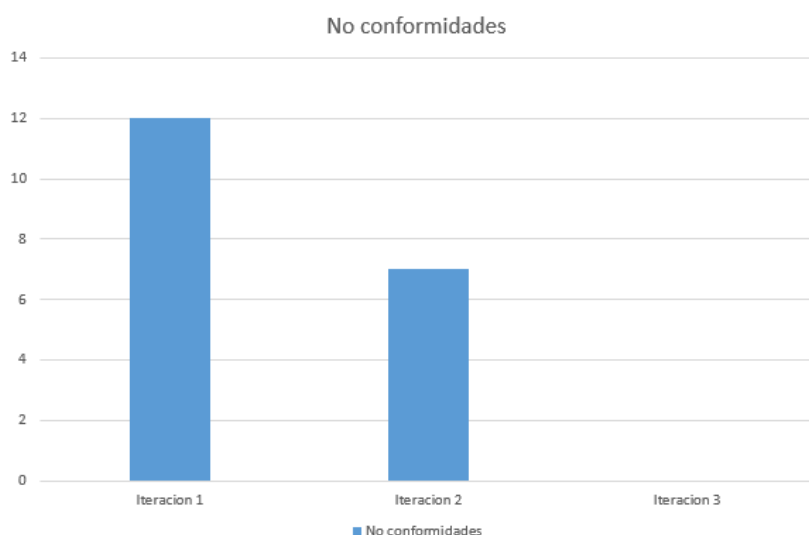


ILUSTRACIÓN 0.7 NO CONFORMIDADES IDENTIFICADAS POR ITERACIONES DE PRUEBA DE USABILIDAD

Las pruebas de seguridad permiten detectar los riesgos a los que está expuesto tu negocio con activos digitales. Riesgos que se convierten en graves vulnerabilidades que permiten a los atacantes robar datos confidenciales, realizar transacciones fraudulentas o tomar el control de la infraestructura operativa del negocio. (QALovers, 2021). Se utiliza como técnica la evaluación heurística.

Luego de la aplicación de las pruebas de seguridad mediante la lista de chequeo se detectan una serie de no conformidades, en dos iteraciones. En una primera iteración se aprecian 7 no conformidades, las cuales fueron solucionadas y en una segunda

iteración no se manifiestan nuevas no conformidades.([ver anexo 2](#))

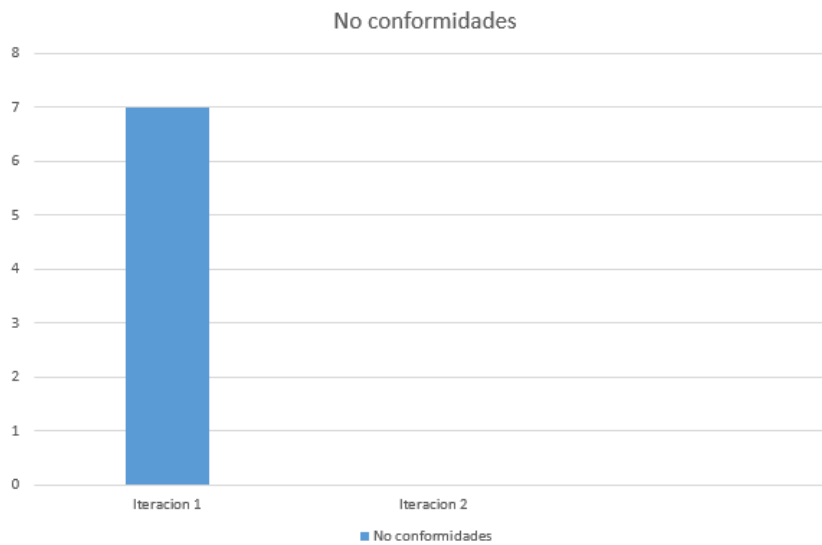


ILUSTRACIÓN 0.8 NO CONFORMIDADES IDENTIFICADAS POR ITERACIONES DE PRUEBA DE SEGURIDAD

3.5 Disciplina 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. (Maida, Esteban Gabriel & Pacienza, Julián, 2015)

Pruebas de Aceptación

Estas pruebas se llevan a cabo por los usuarios finales, quienes establecen una relación directa con las aplicaciones o sistemas desarrollados. Esto permite conocer el desempeño del producto en escenarios reales o muy cercanos a lo que sucederá en el uso constante. Con su empleo verifican si realmente satisfacen las necesidades y requisitos de los clientes. (Tamushi , 2022)

Mediante la aplicación del tipo de prueba Alfa, la cual se realizó con varios usuarios donde todos interactuaron con el sistema probando cada una de las funcionalidades, se mostró que el sistema no presentó ningún tipo de error.

3.6 Conclusiones parciales

En el capítulo se plasmó las pruebas propuestas para la comprobación de los requisitos. Al finalizar con la fase de prueba se obtendría como resultado una aplicación que

funcione correctamente, cumpliéndose así, los objetivos planteados para una mejor satisfacción de su uso para el usuario.

Conclusiones generales

Al concluir con la investigación para el desarrollo del módulo Trámite Docente 1.1 del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED, se pudo arribar a las siguientes conclusiones:

- A partir del análisis de los diferentes tópicos de la investigación, se pudo comprender mejor lo referente a los trámites docentes.
- El empleo de la metodología de desarrollo de software Proceso Unificado Ágil en su versión para la Universidad de las Ciencias Informáticas (AUP-UCI) permitió que se adquirieran los artefactos necesarios del proceso ingenieril, facilitando una mejor comprensión de la solución perfeccionada, mediante la utilización de las herramientas y tecnologías manejadas para su desarrollo.
- Se comprobó la calidad del producto mediante la estrategia de pruebas aplicada y se corroboró que satisface las necesidades del cliente, pudiendo identificar y corregir los errores

Recomendaciones

- Continuar el estudio de la gestión de trámites docentes con el objetivo de optimizar dicho proceso.
- Que el módulo junto con todo el sistema sea utilizado en todos los centros educativos del país.

Referencias bibliográficas

- (s.f.). Obtenido de Ministerio de Educación Superior: <https://www.mes.gob.cu/ministerio>
- Amparo Averbuj. (7 de Marzo de 2022). *Abstracta*. Obtenido de <https://cl.abstracta.us/blog/guia-crear-estrategia-pruebas-software-adecuada/>
- Fernán García de Zúñiga. (3 de febrero de 2020). *arsys*. Obtenido de <https://www.arsys.es/blog/arquitectura-software>
- Santander Universidades . (21 de diciembre de 2020). *Santander Becas*. Obtenido de <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>
- Shelley Pursell. (2021). *Hubspot*. Obtenido de <https://blog.hubspot.es/marketing/pruebas-usabilidad>
- (2021). *¿Qué es un sistema de gestión y para qué sirve?*
- Camilo Clavijo. (2022). *Hubspot*. Obtenido de <https://blog.hubspot.es/sales/modelo-negocio>
- Carla Gomez. (20 de octubre de 2020). *Diario de QA*. Obtenido de <https://www.diariodeqa.com/post/niveles-de-prueba>
- Carlos Lucena Herrera. (14 de Junio de 2019). *OpenWebinars*. Obtenido de <https://openwebinars.net/blog/que-es-prueba-de-rendimiento-software/>
- Cenacad. (14 de enero de 2014). *Censo académico*. Obtenido de <https://cenacad.espol.edu.ec/>
- Comisión Nacional de Acreditación CNA-Chile*. (26 de septiembre de 2018). Obtenido de <https://www.cnachile.cl/noticias/Paginas/GESTI%C3%93N-ACAD%C3%89MICA.aspx>
- Consejos técnicos. (2020 de octubre de 2020). *LoadView.com*. Obtenido de <https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/>
- Desarrollo Web. (7 de febrero de 2020). *Desarrollo Web*. Obtenido de <https://desarrolloweb.com/>: <https://desarrolloweb.com/articulos/181.php>
- Díaz-Barriga, Á. (2013). *tic en el trabajo del aula. Impacto en la planeación. Revista Iberoamericana de Educación Superior*, 4.

Django REST framework. (s.f.). Obtenido de https://www-django--rest--framework-org.translate.google/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es-419&_x_tr_pto=sc#license

Ebooks Online. (9 de mayo de 2021). Obtenido de https://ebooksonline.es/que-es-una-prueba-del-sistema-tipos-y-definicion-con-ejemplo/#%C2%BFQue_es_una_prueba_del_sistema

Eduardo Salazar Martínez. (1 de enero de 2015). *Procedimiento para realizar pruebas de Caja Blanca*. Obtenido de <https://www.informatica-juridica.com/trabajos/procedimiento-realizar-pruebas-caja-blanca/>

Ekon. (28 de julio de 2021). Obtenido de <https://ekon.es/blog/sistemas-de-gestion-integral-para-el-funcionamiento-optimo-de-la-empresa/>

ESAN, C. (30 de septiembre de 2015). *Resistencia al cambio: ¿Cómo enfrentarla?* Obtenido de <https://www.esan.edu.pe/conexion-esan/resistencia-cambio-como-enfrentarla>

Escobar Lascarro, Jesús David. (2019). *Repositorio Institucional*. Obtenido de <https://repositorio.unicartagena.edu.co/handle/11227/13572>

García Pérez, Carlos Rubén. (junio de 2018). Obtenido de <https://repositorio.uci.cu/handle/123456789/9892>

Git. (s.f.). Obtenido de https://git--scm-com.translate.google/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es-419&_x_tr_pto=sc

Grisel Infante Costa. (2007). *Universidad de las Ciencias Informáticas*. Obtenido de <https://publicaciones.uci.cu/index.php/serie/article/view/253>

Guevara, A. D. (2011). *ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA LA GESTIÓN ACADÉMICA DE UN INSTITUTO SUPERIOR TECNOLÓGICO*. Lima, Perú.

Herrera, C. L. (2019). *openwebinars*. Obtenido de <https://openwebinars.net/blog/que-es-prueba-de-rendimiento-software/>

Huihui Liu; Xufang Gong; Li Liao; Bixin Li. (2018). *IEEEEXPLORE*. Obtenido de <https://ieeexplore.ieee.org/abstract/document/8377960>

Jorge Turrado. (10 de marzo de 2020). *campusMVP.es*. Obtenido de <https://www.campusmvp.es/recursos/post/que-son-las-pruebas-de-software.aspx>

Lapiente, M. J. (27 de diciembre de 2019). *Lenguajes hipertextuales: HTML*. Obtenido de <http://www.hipertexto.info>: <http://www.hipertexto.info/documentos/html.htm>

Maida, Esteban Gabriel & Pacienza, Julián. (diciembre de 2015). *METODOLOGIAS DE DESARROLLO DE SOFTWARE*. Obtenido de <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>

Manuel José Gonçalves . (23 de octubre de 2021). *hiberus blog*. Obtenido de <https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/>

MDN contributors. (25 de octubre de 2022). *mdn*. Obtenido de <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>

Miriam Martínez Canelo . (24 de junio de 2020). *Profile*. Obtenido de https://profile.es/blog/patrones-de-diseno-de-software/#%C2%BFQue_son_los_patrones_de_diseno_design_patterns

Molina., O. E. (24 de enero de 2018). *alfabetizacion digital*. Obtenido de <https://www.alfabetizaciondigital.redem.org/potenciando-el-intercambio-acerca-de-las-tics-y-la-educacion-en-cuba/>

Monzó, M. (13 de julio de 2016). *El impacto de las TIC en la sociedad actual*. Obtenido de <https://todoturiam.com/el-impacto-de-las-tic-en-la-sociedad-actual/>

OpenWebinars. (22 de 7 de 2022). Obtenido de <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>

PMOinformatica. (6 de febrero de 2017). *PMOinformatica*. Obtenido de <http://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html>

PostgreSQL. (25 de febrero de 2018). *PostgreSQL*. Obtenido de <http://www.postgresql.org.es>: http://www.postgresql.org.es/sobre_postgresql

Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico*. McGraw-Hill.

Pressman, R. S., & Maxim, B. R. (2015). *Software engineering: a practitioner's approach*. New York: McGraw-Hill Education.

QALovers. (12 de abril de 2021). Obtenido de <https://www.qalovers.com/2019/01/que-son-las-pruebas.html>

Robledano, A. (23 de septiembre de 2019). *OpenWebinars*. Obtenido de OpenWebinars: <https://openwebinars.net/blog/que-es-python/>

Rosario Peiró. (29 de septiembre de 2020). *economipedia*. Obtenido de <https://economipedia.com/definiciones/informacion-2.html>

Santander Universidades. (9 de abril de 2021). *Santander Becas*. Obtenido de <https://www.becas-santander.com/es/blog/python-que-es.html>

Talent Garden. (17 de septiembre de 2021). Obtenido de <https://talentgarden.org/es/data/the-difference-between-conceptual-and-logical-data-model/>

Tamushi . (23 de agosto de 2022). *Testing it*. Obtenido de <https://www.testingit.com.mx/blog/pruebas-de-aceptacion-de-software>

Universidad de las Ciencias Informáticas . (s.f.). Obtenido de <https://www.uci.cu/investigacion-y-desarrollo/productos/xedro/gespro-1305>

Universidad de las Ciencias Informáticas . (s.f.). Obtenido de <https://www.uci.cu/investigacion-y-desarrollo/productos/xabal/excriba>

Universidad de las Ciencias Informáticas . (2015). *Metodología de desarrollo para la Actividad productiva de la UCI*. . La Habana. Cuba .

Universidad de las Ciencias Informáticas. (2014). *Sistema de Gestión Académica de Postgrado versión 2.0*. La Habana : Repositorio UCI.

Universidad de las Ciencias Informáticas. (s.f.). *Universidad de las Ciencias Informáticas*. Obtenido de <https://www.uci.cu/ingenieria-de-requisitos>

Visual Paradigm. (s.f.). Obtenido de Visual Paradigm: <https://circle.visual-paradigm.com/docs/introduction-and-system-requirements/introduction-of-visual-paradigm/>

W3Schools. (2020). *W3Schools*. Obtenido de <https://www.w3schools.com/js/>

William Mc Mullin. (27 de mayo de 2021). *Parasoft*. Obtenido de <https://es.parasoft.com/blog/how-to-write-test-cases-for-software-examples-tutorial/>

WordReference.com. (s.f.). Obtenido de *WordReference.com*: <https://www.wordreference.com/definicion/tr%C3%A1mite>

WordReference.com. (s.f.). Obtenido de *WordReference.com*:
<https://www.wordreference.com/definicion/docente>

Anexos

Anexo 1 : Lista de chequeo para la prueba de usabilidad

	1. ¿Se ofrecen valores por defecto en los campos en caso de que tenga duda?	Si
	2. ¿Es capaz de completar el trabajo rápidamente utilizando este sistema?	Si
	3. ¿La información que proporciona el sistema es efectiva ayudando a completar las tareas?	Si
	4. ¿El sistema tuvo todas las herramientas que se esperaba que tuviera?	No
	5. ¿El sitio tiene una URL, clara y fácil de recordar?	Si
	6. ¿Todos los términos utilizados en el menú fueron comprensibles?	Si
	7. ¿La información que se presenta en la aplicación es fácil de entender y memorizar?	No
	8. ¿Comprendió cómo navegar en la aplicación con facilidad?	No
	9. ¿Encontró la opción deseada en el menú fácilmente?	Si

	10. ¿Fue fácil aprender a utilizar este sistema?	Si
	11. ¿Cuándo realizó una operación con más de un paso y continuó para el siguiente, pudo regresar atrás para consultar la información anterior?	Si
	12. ¿Es fácil volver a encontrar información previa?	Si
	13. ¿Al dar clic en el botón "Atrás" siempre lleva de vuelta a la página de dónde vino?	Si
	14. ¿Las páginas de la aplicación demoraron mucho en cargar?	No
	15. ¿Pudo ver la aplicación en el navegador sin problemas?	Si
	16. ¿Necesitó volver a escribir la información solicitada en páginas anteriores?	Si
	17. ¿En algún momento se sintió perdido en la aplicación, no sabía en qué página se encontraba?	No
	18. ¿Cuándo se selecciona un ícono se diferencia de los no seleccionados?	Si
	19. ¿Pudo acceder a la página de inicio en todo momento?	Si

	20. ¿Encontró el acceso a la ayuda con facilidad?	Si
	21. ¿Necesitó utilizar la ayuda para completar operaciones?	Si
	22. ¿En situaciones donde se pueden producir errores de escritura existe la posibilidad de seleccionar la información de una lista?	Si
	23. ¿Se mostraron mensajes de error para indicarle problemas?	Si
	24. ¿Si cometió algún error al escribir en los campos la aplicación le mostró mensaje de error?	Si
	25. ¿El mensaje de error informa cuales son las acciones a realizar para arreglarlo?	Si
	26. ¿El mensaje de error permite volver a donde se encontraba trabajando?	Si
	27. ¿Pudo corregir los errores fácilmente?	Si
	28. ¿Observó cambios de letra en el sitio web?	No
	29. ¿Presentó problemas con el tamaño de letra para leer la información?	Si

	30. ¿En los listados tiene la opción de seleccionar la cantidad de elementos que desea ver por página?	No
	31. ¿Le fue necesario utilizar la barra de desplazamiento horizontal para desplazarse por el sitio?	No
	32. ¿Las cajas de texto en los formularios tienen el tamaño adecuado para el dato que se debe introducir?	No
	33. ¿Pudo aumentar y disminuir el tamaño de letra?	No
	34. ¿Todos los textos se visualizan en el navegador sin problemas?	No
	35. ¿La presentación se visualiza sin problemas en el navegador?	Si
	36. ¿Los formularios funcionan sin problemas en el navegador?	Si
	37. ¿Ha identificado errores ortográficos?	No
	38. ¿La página refleja la identidad del centro donde fue desarrollada (logos, ...)?	Si

	39. ¿La interfaz de búsqueda está ubicada en la parte superior derecha de la página?	No
	40. ¿El logo de la organización está ubicado en el mismo lugar en todas las páginas, y hacer clic en el logo retorna al usuario a la página de inicio?	Si
	41. ¿El color del texto y del fondo son muy parecido?	Si
	1. ¿Se proporciona un texto equivalente para todo elemento no textual, tales como imágenes, para explicar su contenido a discapacitados visuales?	No
	2. ¿Se han creado atajos de teclado?	Si
	3. ¿Todos los enlaces son descriptivos (para usuarios no videntes)?	Si
	4. ¿Si se incrementa el tamaño de la letra el esquema del sitio se quiebra? Cambiando el tamaño de letra del navegador. Ejemplo en Mozilla sería Herramienta/Opciones/Contenido.	No
	5. ¿El sitio funciona correctamente en un lector de	No

	pantalla y/o navegador de voz?	
	6. ¿Es posible la navegación sin ratón?	Si
	7. ¿La información transmitida a través de los colores también está disponible sin color?	No
	8. ¿Existen textos aclaratorios sobre imágenes de forma que puedan ser comprendidas por cualquier persona independientemente de la discapacidad?	No
	9. ¿Existe un enlace a una página accesible si la consultada no lo es?	No

Anexo 2: Lista de chequeo de la prueba de seguridad


No	Pregunta	Respuesta
1	¿Existen cuentas con contraseñas en blanco?	Si
2	¿Se muestran a los usuarios errores específicos de validación como resultado de un acceso fallido?	Si
3	¿Puede identificar una respuesta diferente de la aplicación cuando se utiliza un usuario válido y una contraseña incorrecta?	Si
4	¿Existen cuentas con contraseña igual al usuario?	NO

5	¿Las contraseñas están validadas para que sean alfanuméricas, y que incluyan caracteres especiales y mayúsculas?	SI
6	¿Las contraseñas están validadas para que tengan seis caracteres como mínimo de longitud?	No
7	¿Las contraseñas y los usuarios son adivinables?	No
8	¿Se bloquea la cuenta del usuario después de cierto número de intentos de acceso fallidos?	No
9	¿El sistema permite a los usuarios cambiar su contraseña?	SI
10	¿El sistema controla el historial de contraseña para prever que no se repitan? (cuando haya transcurrido un tiempo o hasta que no se hayan utilizado otras X cantidad de contraseñas diferentes)	No
11	¿Se puede acceder directamente a una página protegida a través de la barra de direcciones en tu navegador?	No
12	¿El sistema cierra sesión cuando transcurrió un lapso de tiempo X de inactividad?	No
13	¿Es posible acceder a funciones y recursos que deben ser accesibles solo a usuarios que tengan un perfil o privilegios diferentes?	No
14	¿El sistema cifra los identificadores de sesión?	SI
15	¿Un usuario estándar (no administrador) puede modificar sus privilegios en el sistema o los de otros usuarios?	NO
16	¿El sistema actualiza inmediatamente la gestión que se realizó sobre los usuarios incluyendo los roles y permisos?	SI
17	¿Después de una autenticación exitosa si se copia la URL y se cierra sesión es posible acceder nuevamente con esa URL a la aplicación?	Si
18	¿Al cerrar sesión, si se presiona el botón "Atrás" del navegador es posible acceder nuevamente a la aplicación?	NO
19	¿Existe un botón de cierre de sesión visible y que esté presente en todas las páginas que requieren autenticación?	No
20	¿Es posible acceder a un recurso protegido, aunque el usuario no este autenticado?	NO

Anexo 3: Prototipo de interfaz. Registrar trámite de matrícula.

Registrar trámite de matrícula


1 Información del trámite ————— 3 Tutores



Nombres y apellidos: Magela Crespo Revol
Carné de identidad: 15022201613
Sexo: Femenino

Centro educacional: Pedro Téllez Valdés
Especialidad: Primaria
Año académico: Primer grado

Datos de matrícula

Curso escolar: Seleccione v	Fecha de matrícula: Seleccione fecha 	Número de matrícula: 00 - 000
Tomo: <input type="text"/>	Folio: <input type="text"/>	Modalidad de estudio: Seleccione v
Tipo de curso: Seleccione v	Plan de estudio: Seleccione v	Versión de plan de estudio: Seleccione v
Organización de plan de estudio: Seleccione v	Año académico: Seleccione v	Periodo lectivo: Seleccione v
Centro educacional: Seleccione v	Unidad organizativa: Seleccione v	Vía de ingreso: Seleccione v
Situación escolar: Seleccione v	Sesión: Seleccione v	

Datos de estudiante

Enfermedad o padecimiento: Seleccione v	Discapacidad: Seleccione v	Estatura (cm): <input type="text"/>
Peso (kg): <input type="text"/>	Condición de matrícula: Seleccione v	Huérfano: Seleccione v

Includido

Anexo 4: Prototipo de interfaz. Ver detalles de trámite de traslado.

Detalles del trámite de traslado

Estudiante




	Primer nombre: Magela	Carné de identidad: 93032201613	Tramitador: Pedro Tellez Valdés
	Segundo nombre:	Sexo: Femenino	Fecha de inicio del trámite: 10-09-2021
	Primer apellido: Crespo	Situación escolar: Promovido	Estado del trámite: En proceso
	Segundo apellido: Revol	Curso escolar: 2020-2021	

Datos del trámite

Fecha de traslado: 15-09-2021	Fecha de la provincia: 2020-2021	Provincia de destino: Pinar del Río
Orden del centro: Roberto Torres Perez	Orden de la provincia: Pedro Torres	Municipio de destino: La Palma
Fecha del municipio: 03-09-2021	Aprobado por la provincia: Maria Suarez	Centro:
Orden del municipio: Olga Díaz Hernandez	Motivo: Cambio de dirección	
Aprobado por el municipio: Juan Perez		


[Cerrar](#)

Anexo 5: Prototipo de interfaz. Registrar trámite de ubicación por registro

admin 

Registrar trámite de ubicación por registro

Información del trámite

	Nombres y apellidos: Magela Crespo Revol
	Carné de identidad: 15022201613
	Sexo: Femenino

Información de destino

Nivel educativo: Seleccione v	Educación: Seleccione v	Tipificación de la educación: Seleccione v
Especialidad: Seleccione v	Tipo de centro: Seleccione v	Centro educacional: Seleccione v
Curso escolar: Seleccione v	Año académico: Seleccione v	Situación escolar: Seleccione v

[Cancelar](#) [Aceptar](#)

Universidad de las Ciencias Informáticas. XAUCE, Akademos. © 2020-2022

Anexo 6: Prototipo de interfaz. Listar estudiantes

admin

Estudiantes

Nombre y Apellidos, Carné de identidad, Tomo RMG, Folio RMG, Número de matrícula

Foto	Nombre y apellidos	Carné de identidad	Opciones
	José González Delgado	07051236598	
	Pablo Hernández del Sol	00122596587	
	Pedro Gutiérrez Ramirez	15093158423	
	Miguel Rodríguez Pérez	97060332654	

Mostrando 1 a 10 de 4 registros