



Facultad de Ciencias y Tecnologías Computacionales

# Aplicación de escritorio del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED

---

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

**Autor:**

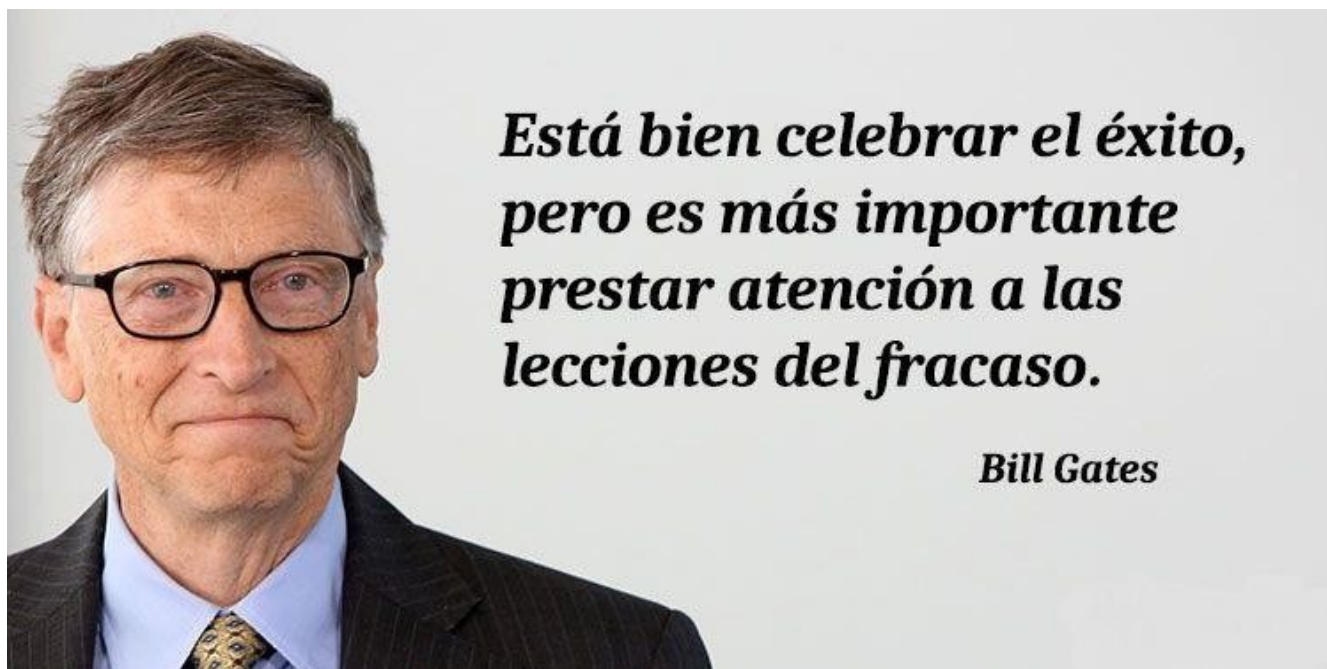
Yesenia Alfonso Ramirez

**Tutores:**

Ing. Néstor Llerena Rivera  
Ing. Rosmery Pedraza Ceballo

La Habana, junio de 2022

"Año 63 de la Revolución"



**DECLARACIÓN DE AUTORÍA**

El autor del trabajo de diploma con título **“Aplicación de escritorio 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 declara como único autor de su contenido. Para que así conste firma(n) la presente a los <día> días del mes de <mes> del año <año>.

**<nombre del autor>**

\_\_\_\_\_  
Firma del Autor

**<nombre del tutor>**

\_\_\_\_\_  
Firma del Tutor

**<nombre del autor>**

\_\_\_\_\_  
Firma del Autor

**<nombre del tutor>**

\_\_\_\_\_  
Firma del Tutor

## **DATOS DE CONTACTO**

Nombre y apellidos: Néstor Llerena Rivera,

Títulos académicos: Ingeniería en Ciencias Informáticas,

Correo electrónico: nllerena@uci.cu

Nombre y apellidos: Rosmery Pedraza Ceballo

Títulos académicos: Ingeniería en Ciencias Informáticas,

Correo electrónico: rceballo@uci.cu

## **AGRADECIMIENTOS**

Agradecer, significa sentir gratitud, sentimiento que nos obliga a estimar el beneficio o favor que se nos ha hecho y corresponder a él.

Mis primeras palabras de agradecimiento van dirigidas a mi Compañero por excelencia, quien no me ha dejado ni un segundo, el que me ha dado el aliento de vida, quien ha puesto combustible al motor de mi existencia, quien me ha coronado de favores sin merecerlo, el que me levanta cuando caigo, al que sana mis heridas y al que me dio el conocimiento para poder llegar a este punto de mi carrera, a mi Dios, por el cual vivo.

A mi mamá, Marbelis, mi ejemplo de mujer virtuosa, que junto a mí pasó madrugadas, consintiéndome y apoyándome en todo cuanto me hiciera falta para continuar adelante en los estudios.

A mi papá bello, Jorge Luis, porque no ha dejado de confiar en mí, siempre supo que en algún punto le iba a devolver tantos años de sacrificio con este título.

A mi amado esposo, Michel, quien ha sido súper paciente este último año, lleno de pruebas, de frustración, en ocasiones con ganas de tirar los guantes; gracias le doy porque ha sido mi confidente y más que eso quién levantó mis brazos cuando mis fuerzas no daban más.

A mi abuela, Neris, que se fue al cielo justo en este proceso de tesis. Quizás para ella no era tan importante este título porque se fijaba más en mis cualidades como persona, sus ojos no estaban en mi intelecto sino en mi corazón, pero durante el tiempo que vivió me hizo saber lo orgullosa que estaba de mí por haber llegado tan lejos, gracias mima.

A mis pastores, Yamil y Kenia, por sus oraciones, sin lo cual no hubiera podido seguir.

A mi amiga Claudia que me regaló tantos momentos de distracción en medio del estrés y con quien siempre he podido contar.

A dos personitas incondicionales que han sido el manual difícil de encontrar pero que tiene todas las respuestas, Dayana y Javier, muchísimas gracias por todo.

A mis tutores, Rosmery y Néstor, por la paciencia que tuvieron y por ayudarme en todo.

Por último, pero no menos importante, a mi familia por ser especial y hermosa.

Les agradezco con todo mi corazón.

## **DEDICATORIA**

A mis padres por los cuáles cursé la universidad.

A mi esposo porque me ha hecho ser perseverante.

A Dios.

## RESUMEN

Los sistemas de gestión académica permiten digitalizar la información referente a todo el proceso educativo del estudiante, ya sean, calificaciones, tareas, comunicados, reuniones, horarios de clases, fechas de exámenes y otros eventos, contribuyendo a la simplificación del trabajo del personal administrativo y al procesamiento de grandes cantidades de datos. Dada su gran importancia en el proceso académico, la Universidad de Ciencias Informáticas (UCI), en el Centro de Tecnologías para la Formación (FORTES) desarrolla el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED, sistema que tiene como objetivo el control del recorrido de cada estudiante del país desde la enseñanza primaria hasta la conclusión de estudios universitarios. Para que esto ocurra es necesario que el sistema funcione en cada escuela del país, aún si en estas no existe conexión a internet, por lo que surge la necesidad de crear una solución que permita el uso del Sistema de Gestión Académica XAUCE AKADEMOS sin conexión a internet que permita importar y exportar información al sistema académico web visible en todo el país. El sistema fue desarrollado en el framework Electron bajo la arquitectura de microservicios.

**Palabras clave:** Sistema, Gestión Académica, sin conexión a internet.

## **ABSTRACT**

*The academic management systems allow the digitalization of information related to the entire educational process of the student, whether grades, assignments, communications, meetings, class schedules, exam dates and other events, contributing to the simplification of the work of administrative staff and the processing of large amounts of data. Given its great importance in the academic process, the Universidad de Ciencias Informáticas (UCI), in the Centro de Tecnologías para la Formación (FORTES) develops the Academic Management System XAUCE AKADEMOS for MINED, a system that aims to control the path of each student in the country from elementary school to the completion of university studies. For this to happen, it is necessary that the system works in every school in the country, even if there is no internet connection, so the need arises to create a solution that allows the use of the Academic Management System XAUCE AKADEMOS without internet connection that allows importing and exporting information to the web academic system visible throughout the country. The system was developed in the Electron framework under the microservices architecture.*

**Keywords:** *System, Academic Management, offline.*

## TABLA DE CONTENIDOS

<u>INTRODUCCIÓN</u> .....	1
<u>CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA</u> .....	5
<u>1.1 Marco conceptual</u> .....	5
<u>1.1.1 Aplicaciones o software de escritorio sin conexión a internet</u> .....	5
<u>1.2 Aplicaciones de escritorio sin conexión a internet para Sistemas de Gestión Académica</u> .....	6
<u>1.2.1 Sistema</u> .....	6
<u>1.2.2 Gestión Académica</u> .....	7
<u>1.2.3 Definición de Sistema de Gestión Académica</u> .....	7
<u>1.2.4 Definición de Aplicación de escritorio sin conexión a internet para Sistema de Gestión Académica</u> .....	8
<u>1.3 Análisis de sistemas homólogos</u> .....	8
<u>1.3.1 Sistema de Gestión Académica (SGA)</u> .....	8
<u>1.3.2 Sistema de Gestión Académica (aGora)</u> .....	8
<u>1.3.2 Sistema de Gestión Académica (CLASS)</u> .....	8
<u>1.4 Metodología de Desarrollo de Software</u> .....	10
<u>1.4.1 Metodología AUP-UCI</u> .....	11
<u>1.4.3 Fases de AUP-UCI</u> .....	12
<u>1.4.4 Disciplinas de AUP-UCI</u> .....	12
<u>1.5 Herramientas y tecnologías</u> .....	13
<u>1.5.1 Lenguajes de Programación</u> .....	14
<u>1.5.2 Lenguajes de Marcado de Hipertexto</u> .....	14
<u>1.5.3 Entorno de Desarrollo Integrado (IDE)</u> .....	14
<u>1.5.4 Marco de Trabajo</u> .....	15
<u>1.5.5 Lenguaje Unificado de Modelado (UML)</u> .....	15
<u>1.5.6 Herramienta de Modelado</u> .....	15
<u>1.6 Conclusiones del capítulo</u> .....	16
<u>CAPÍTULO 2: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO</u> .....	17
<u>2.1 Modelado del dominio</u> .....	17
<u>2.1.1 Diagrama de proceso de negocio</u> .....	17
<u>2.1.2 Modelo Conceptual</u> .....	19
<u>2.2 Requisitos del sistema</u> .....	19
<u>2.2.1 Requisitos Funcionales:</u> .....	20
<u>2.2.2 Requisitos no funcionales:</u> .....	21
<u>2.2.3 Descripción de requisitos por procesos</u> .....	22
<u>2.3 Diseño</u> .....	23
<u>2.3.1 Modelo de Datos</u> .....	23
<u>2.4 Arquitectura del Sistema</u> .....	24
<u>2.5 Patrones de diseño</u> .....	25



2.5.1 Modelo-Vista-Modelo de Vista.....	26
2.5.2 Patrones GRASP.....	27
Conclusiones del capítulo.....	27
<b>CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....</b>	<b>28</b>
3.1 Estándar de codificación.....	28
3.1.2 Diagrama de componentes.....	31
3.3 Pruebas de software.....	32
3.3.1 Pruebas a realizar.....	32
3.3.2 Método de prueba.....	33
3.3.3 Técnica de prueba.....	34
3.4 Aplicación de las pruebas.....	35
3.4.1 Pruebas internas.....	35
3.4.2 Pruebas de Aceptación.....	41
3.5 Evaluación del objetivo general de la investigación.....	41
Conclusiones parciales.....	43
<b>CONCLUSIONES FINALES.....</b>	<b>45</b>
<b>RECOMENDACIONES.....</b>	<b>46</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>47</b>
<b>ANEXOS.....</b>	<b>50</b>
Anexo 1: Descripción de Requisitos por procesos.....	50
Anexo 2: Acta de Aceptación del Producto.....	61
Anexo 3: Encuesta inicial de la investigación.....	61

## ÍNDICE DE TABLAS

<b>Tabla 1:</b> Comparación entre Sistemas Homólogos y el Sistema de Gestión Académica XAUCE AKADEMOS.....	9
<b>Tabla 2:</b> Comparación entre Metodologías Tradicionales y Metodologías Ágiles. Tomado de: (Figuroa et al., 2008).....	10
<b>Tabla 3:</b> Especificación de los requisitos funcionales: Registrar trámite de registro, modificar trámite de registro, Listar trámite de registro y Ver datos de un trámite de registro.....	21
<b>Tabla 4:</b> Descripción textual del requisito "Ver datos de un trámite de registro".....	22
<b>Tabla 5:</b> Listado de caminos independientes.....	37
<b>Tabla 6:</b> Caso de prueba de unidad para el camino 1.....	38
<b>Tabla 7:</b> Caso de prueba de unidad para el camino 2.....	38
<b>Tabla 8:</b> Caso de prueba de unidad para el camino 3.....	38
<b>Tabla 9:</b> Descripción de las variables del caso de prueba "Es electron".....	39
<b>Tabla 10:</b> Descripción de los escenarios del caso de prueba " Configuración del guard".....	40
<b>Tabla 11:</b> Cuadro lógico de IADOV.....	42

## ÍNDICE DE FIGURAS

<b>Figura 1:</b> Diagrama del proceso de negocio “Ver datos de un trámite de registro”.....	18
<b>Figura 2:</b> Modelo Conceptual.....	19
<b>Figura 3:</b> Modelo de Datos.....	24
<b>Figura 4:</b> Arquitectura basada en el estilo arquitectural En Capas.....	25
<b>Figura 5:</b> Identación.....	28
<b>Figura 6:</b> Tabuladores y espacios.....	28
<b>Figura 7:</b> Importaciones. Elaboración propia.....	28
<b>Figura 8:</b> Declaraciones de variables.....	29
<b>Figura 9:</b> Tamaño máximo de líneas.....	29
<b>Figura 10:</b> Comentarios.....	29
<b>Figura 11:</b> Asignación de nombres simples.....	30
<b>Figura 12:</b> Asignación de nombres compuestos.....	30
<b>Figura 13:</b> Líneas en blanco.....	31
<b>Figura 14:</b> Diagrama de Componentes de la funcionalidad genérica Crear.....	31
<b>Figura 15:</b> Diagrama de despliegue.....	32
<b>Figura 16:</b> Esquema del Método de Caja Negra.....	34
<b>Figura 17:</b> Enumeración de las líneas de código.....	36
<b>Figura 18:</b> Grafo de flujo.....	37
<b>Figura 19:</b> Estadísticas de las pruebas.....	41
<b>Figura 20:</b> Ubicación del índice de satisfacción Grupal.....	43

## **INTRODUCCIÓN**

Con el pasar de los años las tecnologías han evolucionado por lo que el hombre se ha visto en la imperiosa necesidad de crear y emplear masivamente las Tecnologías de la Información y las Comunicaciones (TIC), con el objetivo de mejorar sus condiciones de vida y de trabajo. En la rama educacional la informatización de la gestión académica ha permitido registrar, organizar y administrar las actividades del proceso docente, entre las que se encuentran la gestión del plan de estudio, de matrícula, del control de asistencia y las evaluaciones de los estudiantes.

La gestión académica es el ámbito de acción orientado a facilitar y mejorar los procesos formativos que imparten las instituciones de educación ("Formación en Gestión Académica," 2018). En el mundo se han creado varios sistemas que están orientados a gestionar todos los procesos académicos, entre ellos, el Sistema de Gestión de Información Académica de la Universidad Distrital Francisco José de Caldas (Caldas, 2021), de la Universidad Pública de Colombia y el Sistema de Gestión Académica de la Universidad Iberoamericana Ciudad de México (México, 2022).

Los sistemas de gestión académica permiten digitalizar la información referente a todo el proceso educativo del estudiante, ya sean, calificaciones, tareas, comunicados, reuniones, horarios de clases, fechas de exámenes y otros eventos, contribuyendo a la simplificación del trabajo del personal administrativo y al procesamiento de grandes cantidades de datos. Dada la gran importancia de estos sistemas se extendieron a plataformas de escritorio. Estas aplicaciones tienen como ventaja que no requieren de comunicación con el exterior repercutiendo en una mayor velocidad de procesamiento, además su tiempo de respuesta es muy rápido (Gutierrez, 2020).

El Sistema de Gestión Académica (SGA) (Chacon, 2019) en su versión 3.0 es un ejemplo de aplicación de escritorio para un sistema de gestión académica, al igual que CLASS, Sistema de Gestión Académica (SGA) con integración financiera contable (Innovasoft, 2022).

Actualmente Cuba se encuentra asediada por una serie de políticas y leyes de restricción inherentes al bloqueo económico, donde el flujo de información y el acceso más amplio a internet del pueblo cubano es uno de los objetivos de ataque. Entre las restricciones más

notorias está: la prohibición del acceso a determinadas plataformas y tecnologías, el empleo del ciberespacio para intentar subvertir el sistema político, influir en el orden interno del país e intervenir en los esfuerzos en el desarrollo de internet.

Por lo anteriormente planteado Cuba no puede acceder a sistemas de gestión académica ya creados en otros países. Respondiendo a este impedimento y en apoyo a la soberanía tecnológica de nuestro país, la Universidad de las Ciencias Informáticas (UCI), en Centro de Tecnologías para la Formación (FORTES) se desarrolla el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED. Este software permite la gestión del proceso académico a partir de varios módulos, a través de los cuales se controla el proceso de matrícula y evaluaciones, entre otros.

Este sistema permite el control del recorrido de cada estudiante del país desde la enseñanza primaria hasta la conclusión de estudios universitarios. Por ello, es imprescindible que cada escuela de Cuba pueda tener acceso a dicho sistema, pero actualmente sólo se puede usar mediante conexiones a internet. Por consiguiente, surge la necesidad de crear una solución que permita el uso del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED sin conexión a internet que permita importar y exportar información al sistema académico web visible en todo el país lo que constituye la **situación problemática** de la presente investigación.

A partir de la **situación problemática**, se propone como **problema científico**: ¿Cómo garantizar el uso del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED en los centros educacionales del país sin conexión a internet? En correspondencia con el problema planteado, se formula como **objeto de estudio**: Aplicaciones de escritorio sin conexión a internet. Con el fin de dar solución a dicho problema se define como **objetivo general**: Desarrollar una aplicación de escritorio sin conexión a internet que permita el uso del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED, enmarcado en el **campo de acción**: Aplicaciones de escritorio sin conexión a internet para Sistemas de Gestión Académica.

Para el cumplimiento del objetivo general se definieron los **objetivos específicos** que se proponen a continuación:

Elaborar marco teórico sobre el estado del arte de las herramientas empleadas para desarrollar aplicaciones de escritorio que garantice el uso del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED de manera offline.

Implementar una aplicación de escritorio que permita el uso del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED de manera offline.

Validar la solución propuesta aplicando métodos de validación de software.

De acuerdo al objetivo expuesto se definen las siguientes tareas investigativas:

1. Sistematización de los conceptos fundamentales del dominio de la investigación.
2. Realización de un estudio sobre el desarrollo de aplicaciones de escritorio.
3. Selección de las tecnologías a utilizar para el desarrollo de una aplicación de escritorio que permita el uso de manera offline del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED.
4. Aplicación de las fases, disciplinas y etapas de la metodología de desarrollo de software seleccionada.
5. Aplicación de las pruebas de calidad de software.

Luego de obtener toda la información relacionada con el tema de la investigación se plantean como posibles resultados:

1. Aplicación de escritorio que permita el uso del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED de manera offline.
2. Artefactos de Ingeniería de Software para el desarrollo de la solución.

Para dar cumplimiento a las tareas de investigación anteriormente expuestas, se emplean diferentes métodos de la investigación científica, los cuales son la ruta a seguir mediante una serie de procedimientos fijados de antemano, para alcanzar un determinado fin.

#### **Métodos Teóricos:**

- ✓ **Histórico-Lógico:** Su empleo fue dirigido al análisis de la documentación asociada a las aplicaciones de escritorio sin conexión a internet a través de la historia, y en la investigación de aplicaciones similares, para de esta manera poder definir un concepto propio necesario para la implementación de la solución.

- ✓ **Analítico-Sintético:** Se utilizó para realizar un estudio general de las aplicaciones de escritorio sin conexión a internet para posteriormente llegar a una conclusión propia que permita resolver el problema científico.
- ✓ **Modelación:** Posibilitó la elaboración de múltiples diagramas para un mejor entendimiento y solución del problema.

#### **Métodos empíricos:**

- ✓ **Revisión y consulta de la información:** Permitió el estudio de diferentes documentos y tipos de bibliografías, posteriormente se seleccionó la información necesaria de la investigación.
- ✓ **Observación:** Con este método es posible distinguir directamente los hechos de la realidad objetiva. Permite conocer el proceso delimitado como objeto de estudio, lo cual contribuyó a tener registro visual más detallado de lo que se quiere y hace falta hacer.

#### **El presente trabajo de diploma se encuentra estructurado por tres capítulos:**

##### **Capítulo 1:** “Fundamentación teórica”.

En este capítulo se exponen los conceptos fundamentales de las aplicaciones de escritorio offline, así como el estudio de las soluciones existentes en la actualidad. Además, se precisan las herramientas y tecnologías a utilizar para la implementación de la aplicación.

##### **Capítulo 2:** “Análisis y diseño de la aplicación offline para el Sistema de Gestión Académica XAUCE AKADEMOS”

En este capítulo se marcan las pautas, objetivos y restricciones teniendo en cuenta los posibles riesgos a los que se puede enfrentar el desarrollo, lo cual permite crear una base sólida para una buena ejecución de las funcionalidades deseadas. Se realizan descripciones de los requisitos funcionales y no funcionales que se deben tener en cuenta para satisfacer las necesidades del cliente y se define la arquitectura y los patrones de diseño utilizados.

##### **Capítulo 3:** “Implementación y prueba de la aplicación offline para el Sistema de Gestión Académica XAUCE AKADEMOS”

## **CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA**

En este capítulo se realiza una sistematización de los fundamentos teórico-metodológicos asociados al objeto de estudio y al tipo de resultado contenido en el campo de acción, así como el análisis y descripción de las técnicas y herramientas empleadas en los sistemas de gestión académica en su versión de escritorio. Esto conlleva a identificar las principales fuentes bibliográficas, áreas de conocimiento y tecnologías a utilizar.

### **1.1 Marco conceptual**

Para una mejor comprensión de la solución informática a desarrollar, a continuación, se muestran un conjunto de conceptos esenciales:

#### **1.1.1 Aplicaciones o software de escritorio sin conexión a internet**

En los primeros días de los ordenadores, los sistemas operativos, el software que controla las aplicaciones y recursos del sistema, no existían. Tampoco existía el software propiamente dicho. Los ordenadores se programaban conectando manualmente circuitos de cálculo y cambiando el cableado para cada nuevo cálculo a realizar. Esto cambió en 1945, cuando John Von Neumann inventó un ordenador que almacenaba en memoria instrucciones que indicaban a las unidades de cálculo lo que debía hacer. El 21 de junio de 1948 se ejecutó el primer programa en una máquina llamada Baby en la Universidad de Manchester... y nació el software (ComputerWorld, 2003; Stallings, Aguilar, Doderó, Torres, & Mora, 1997).

El software se define como un programa informático o un conjunto de instrucciones para que el ordenador pueda funcionar de forma adecuada; los ordenadores necesitan este software para realizar tareas específicas; el software puede definirse como un término colectivo para las aplicaciones, los programas, los scripts que se ejecutan en un ordenador o en cualquier otro dispositivo programable (Pedamkar, 2019).

Abeer Alkhars y Wasan Mahmoud afirman que una aplicación de escritorio es un software que se ejecuta en un solo ordenador (portátil o de sobremesa) y se utiliza para realizar tareas específicas (Alkhars & Mahmoud, 2017). Por otra parte, la ingeniera en software y sistemas Priya Pedamkar plantea que: el software o las aplicaciones de escritorio pueden considerarse



una versión local de un servicio o de aplicaciones web que proporcionan la opción de interacción entre el procesamiento de la aplicación y el usuario sin conectarse directamente con el servicio web (Pedamkar, 2019). Haciendo referencia también al concepto de aplicaciones de escritorio, en la página V2 Cloud se expone que: una aplicación de escritorio es un programa de software que puede ejecutarse en un ordenador independiente para que el usuario final realice una tarea específica ("What is a Desktop App?," 2020).

Con otros términos, habiendo estudiado los conceptos anteriormente planteados, se puede definir como aplicación o software de escritorio, a la versión local de un servicio o aplicación web que se ejecuta de forma autónoma en un solo ordenador (portátil o de sobremesa), para realizar tareas específicas sin conectarse directamente con el servicio web.

## **1.2 Aplicaciones de escritorio sin conexión a internet para Sistemas de Gestión Académica**

Para una mejor comprensión de la definición de aplicaciones de escritorio sin conexión a internet para Sistemas de Gestión Académica se expusieron varios conceptos de distintos autores.

### **1.2.1 Sistema**

Fuera del mundo de la informática, Rusell L. Ackoff publicó *Systems Thinking and Thinking Systems* (Ackoff, 1994), donde definió un sistema como un conjunto de dos o más partes donde:

- (1) Cada una de las cuales puede afectar al rendimiento o a las propiedades del conjunto.
- (2) Ninguna de las cuales puede tener un efecto independiente sobre el conjunto.
- (3) Ningún subgrupo de las cuales puede tener un efecto independiente sobre el conjunto.

Además Ackoff sostiene que un sistema es un conjunto de partes, algunas de las cuales pueden ser sistemas, que interactúan entre sí (Johnson, 2021).

Desde otro punto de vista, el Instituto de Energías renovables (IER) de la Universidad Nacional Autónoma de México (UNAM), plantea que: sistema, es un conjunto de elementos con relaciones de interacción e interdependencia que le confieren entidad propia al formar un todo unificado ("Instituto de Energías Renovables de la UNAM," 2007).

Por último, pero no menos importante, el 18 de abril del 2019, Dov Dori publica el artículo System Definition, System Worldviews, and Systemness Characteristics (Dori et al., 2019), en el cual explica que un sistema es un arreglo de partes o elementos que juntos exhiben un comportamiento o significado que los constituyentes individuales no tienen.

Después del análisis de las definiciones anteriores se elaboró el siguiente concepto de sistema: Conjunto de elementos con relaciones de interacción e interdependencia que juntos manifiestan un comportamiento o significado que los constituyentes individuales no poseen.

### **1.2.2 Gestión Académica**

De acuerdo con Maria Sonia Andrade Viveros y Luis C. Sanchez Arce en su artículo “The academic management of the socio-critical pedagogical model in the educational institution” (Viveros Andrade & Sanchez Arce, 2018), de la Revista Universidad y Sociedad, la gestión académica son las prácticas recurrentes que permiten a la Institución Educativa asegurar la coherencia de su propuesta curricular con el Proyecto Educativo Institucional (PEI), el marco nacional de la política educativa y las necesidades de aprendizaje de sus estudiantes.

Unos años más tarde Fabricio D. Angulo Luna definió la gestión académica como la aplicación de la teoría y práctica administrativa al campo de la educación, regido por la planificación, organización, dirección, coordinación y control (Angulo Luna, 2021).

Por tanto, la gestión académica es el conjunto de actividades y prácticas recurrentes que aplican la teoría y la práctica administrativa al campo de la educación, regido por la planificación, organización, dirección, coordinación y control.

### **1.2.3 Definición de Sistema de Gestión Académica**

Conjunto de elementos, actividades y prácticas recurrentes con relaciones de interacción e interdependencia que aplican la teoría y la práctica administrativa al campo de la educación, regido por la planificación, organización, dirección, coordinación y control.

#### **1.2.4 Definición de Aplicación de escritorio sin conexión a internet para Sistema de Gestión Académica**

Versión local de una aplicación o servicio web de un Sistema de Gestión Académica que se ejecuta en un solo ordenador (portátil o de sobremesa), para realizar las actividades y prácticas recurrentes con relaciones de interacción e interdependencia que aplican la teoría y la práctica administrativa al campo de la educación. Todo esto sin conectarse directamente con el servicio web.

#### **1.3 Análisis de sistemas homólogos**

En este epígrafe se observan las soluciones existentes de Sistemas de Gestión Académica en plataforma de escritorio y se hace una comparación de ellos con el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED.

##### **1.3.1 Sistema de Gestión Académica (SGA)**

Herramienta que implementa un Sistema de Gestión Académica, es un software que simula la lógica que se implementa en una Matrícula para una Universidad, esto es la creación, modificación, eliminación y la búsqueda de: Estudiantes, Profesores, Ciclos, Grupos, Cursos y Carreras (Chacon, 2019).

##### **1.3.2 Sistema de Gestión Académica (aGora)**

Es un software de gestión completo e integrado que cubre todas las necesidades de cualquier centro docente en términos de planificación, gestión y control académico, comercial, económico, entre otros ("aGora," 2020).

##### **1.3.2 Sistema de Gestión Académica (CLASS)**

Sistema de Gestión Académica (SGA) con integración financiera contable que permite (Innovasoft, 2022):

- ❖ Administrar de forma automatizada, integrada y virtualizada los diferentes procesos administrativos.
- ❖ Compuesto por una robusta suite con más de 50 modulaciones.

❖ Mejora y actualización continua.

**Tabla 1:** Comparación entre Sistemas Homólogos y el Sistema de Gestión Académica XAUCE AKADEMOS

Variables	Sistema de Gestión Académica (SGA)	Sistema de Gestión Académica (CLASS)	Sistema de Gestión Académica (aGora)	Sistema de Gestión Académica (XAUCE AKADEMOS)
Sin conexión a Internet	No	No	Sí	Sí
Licencia	Gratis	Pago	Pago	Gratis
Exportar información	No	Sí	Sí	Sí
Importar información	No	No	Sí	Sí
Multiplataforma	Sí	Sí	Sí	Sí

La comparación realizada anteriormente facilita la comprensión de la necesidad de crear un sistema propio, ya que los softwares implementados actualmente no cumplen con las necesidades del cliente, sin embargo la aplicación desktop del Sistema de Gestión Académica XAUCE AKADEMOS va a funcionar sólo local, permitiendo la importación y exportación de información desde las escuelas donde no existe conexión a internet, para posteriormente añadir esta información al Sistema de Gestión Académica en su versión web visible en todo el país. De esta manera no quedaría ninguna institución educacional del país fuera del sistema, lo que permite llevar el control de cada estudiante desde que comienza sus estudios hasta que culmina.

#### 1.4 Metodología de Desarrollo de Software

El desarrollo del software depende de un sin número de actividades y etapas, donde el impacto de elegir la mejor metodología para un equipo, en un determinado proyecto es trascendental para el éxito del producto. El papel preponderante de las metodologías es sin duda esencial en un proyecto y en el paso inicial, que debe encajar en el equipo, guiar y organizar actividades que conlleven a las metas trazadas en el grupo (Figuroa, Solís, & Cabrera, 2008).

Existen varias clasificaciones para estas metodologías dentro de las cuales se destacan (Figuroa et al., 2008):

- ❖ **Metodologías tradicionales:** Conocidas también como formales, se enfocan en la documentación, planificación y procesos (plantillas, técnicas de administración, revisiones). Otra de las características de este enfoque es el alto costo de realizar un cambio al no ofrecer una buena solución a proyectos donde el entorno es volátil.
- ❖ **Metodologías ágiles:** Utilizadas para dar respuesta a los problemas derivados de las metodologías tradicionales.

Para la presente investigación se decidió utilizar una metodología ágil para aprovechar las facilidades que brinda en cuanto a requerir poco cúmulo de documentación, adaptable fácilmente a un entorno cambiante y cuenta con poco tiempo para el desarrollo; estas facilidades son de gran importancia porque para la implementación del sistema se cuenta con un solo desarrollador, por lo que es más importante una obtención adecuada y funcional del sistema que lo documentado que este pueda estar. Para una mejor comprensión en la Tabla 2 se muestra una comparación entre ambas metodologías.

**Tabla 2:** Comparación entre Metodologías Tradicionales y Metodologías Ágiles. Tomado de: (Figuroa et al., 2008).

<b>Metodologías Tradicionales</b>	<b>Metodologías Ágiles</b>
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo	Basadas en heurísticas provenientes de prácticas de producción de código
Cierta resistencia a los cambios	Especialmente preparados para cambios durante el proyecto

Impuestas externamente	Impuestas internamente (por el equipo)
Proceso mucho más controlado, con numerosas políticas/normas	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del equipo de desarrollo
Más artefactos	Pocos artefactos
Más roles	Pocos roles
Grupos grandes y posiblemente distribuidos	Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio
La arquitectura del software es esencial y se expresa mediante modelos	Menos énfasis en la arquitectura del software
Existe un contrato prefijado	No existe contrato tradicional o al menos es bastante flexible

Dentro de las metodologías de desarrollo ágil como son XP (Extreme Programming), OpenUp (Open Unified Process), Scrum, AUP (Agile Unified Process) se decidió utilizar la metodología AUP versión UCI.

#### 1.4.1 Metodología AUP-UCI

La UCI cuenta con 14 centros productivos que se dedican al desarrollo de software y/o servicios asociados a un dominio de aplicación bien definido. Esta diversidad de centros y proyectos hizo que la actividad productiva en la UCI fuera cada vez más amplia, y trajo consigo la heterogeneidad en el proceso de desarrollo de software (Sánchez, 2015).

A pesar de la variedad de metodologías utilizadas, muy pocos proyectos la aplicaban en su totalidad. Uno de los principales problemas era que sin importar la metodología que se usara se planificaba con un único cronograma tipo, por lo que, en pos de erradicar los problemas existentes, se decidió escoger la metodología Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP), versión simplificada del Proceso Racional Unificado

(RUP), para ser adaptada al ciclo de vida definido para la actividad de la Universidad (Sánchez, 2015).

AUP consta de 4 fases: inicio, elaboración, construcción y transición, además define 7 disciplinas (4 ingenieriles y 3 de gestión de proyectos), modelo, implementación, prueba, despliegue, gestión de configuración, gestión de proyectos y entorno (Sánchez, 2015).

La variación AUP-UCI mantiene la fase de inicio, pero modificando el objetivo de la misma, unifica las fases de elaboración, construcción y transición en una sola fase llamada ejecución y agrega la fase de cierre (Sánchez, 2015).

### **1.4.3 Fases de AUP-UCI**

**Inicio:** Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimados de tiempo, esfuerzo y costo y decir si se ejecuta o no el proyecto (Sánchez, 2015).

**Ejecución:** En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

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

### **1.4.4 Disciplinas de AUP-UCI**

**Modelado de negocio:** El Modelado del Negocio es la disciplina destinada a comprender los procesos del 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 (Sánchez, 2015):

- **Casos de Uso del Negocio (CUN)**
- **Descripción de Proceso de Negocio (DPN)**

➤ **Modelo Conceptual (MC)**

**Requisitos:** Administración y gestión de los requisitos funcionales y no funcionales.

**Análisis y diseño:** En esta disciplina, si se considera necesario, los requisitos pueden ser definidos 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 la arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos que el de análisis (Sánchez, 2015).

**Implementación:** En la implementación, a partir de los resultados del Análisis y diseño se construye el sistema.

**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 posibles componentes de prueba ejecutables para automatizar las pruebas.

**Pruebas de liberación:** Pruebas diseñadas y ejecutadas por una entidad certificada de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.

**Pruebas de Aceptación:** Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

### **1.5 Herramientas y tecnologías**

Las **Herramientas Informáticas** son un conjunto de programas, aplicaciones o, simplemente instrucciones, que al utilizarlas permitirán al usuario realizar un trabajo determinado de la mejor manera posible en algún dispositivo informático (Gonzales Chavez & Andrade Vela, 2022).

Las **Tecnologías Informáticas** pueden definirse como el estudio, diseño, creación, manipulación, asistencia o



gerencia de los sistemas informáticos computarizados que requieren de un Software y un Hardware para funcionar. Su finalidad primordial es facilitar los procesos realizados por el ser humano en búsqueda de comodidad, ahorro y sobre todo, seguridad ("EUROINNOVA (International Online Education)," 2019).

### **1.5.1 Lenguajes de Programación**

**TypeScript:** Es un lenguaje de programación fuertemente tipado que se basa en JavaScript, lo que le brinda mejores herramientas a cualquier escala. Agrega sintaxis adicional a JavaScript para admitir una integración más estrecha con su editor ("typescrip-lang-org," 2022).

TypeScript es un superconjunto de JavaScript desarrollado y mantenido por Microsoft. TypeScript añade tipificación a JavaScript. Debido a que JavaScript no tiene una tipificación estricta, a menudo tiene errores o bugs en el tiempo de ejecución y TypeScript elimina esa posibilidad. Debido a la comprobación estática de TypeScript, éste detecta errores durante el tiempo de compilación. Debido a esta gestión temprana de errores hace que el código base sea más claro y robusto (Koskela, 2022).

Las hojas de estilo en cascada (**CSS**, cascading style sheets) se utilizan para formatear el diseño de las páginas web. Se pueden usar para definir estilos de texto, tamaños de tablas y otros aspectos de las páginas web que anteriormente solo se podían definir en el HTML de una página ("TechTerms," 2007).

### **1.5.2 Lenguajes de Marcado de Hipertexto**

**Lenguaje de marcado de hipertexto (HTML):** Es el código que se utiliza para estructurar y desplegar una página web y sus contenidos. Por ejemplo, sus contenidos podrían ser párrafos, una lista con viñetas, o imágenes y tablas de datos ("Developer.mozilla.org," 2015).

### **1.5.3 Entorno de Desarrollo Integrado (IDE)**

**Visual Studio Code:** Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de

extensiones para otros lenguajes y tiempos de ejecución (como C++, C#, Java, Python, PHP, Go, .NET) ("Code.visualstudio.com," 2022).

#### 1.5.4 Marco de Trabajo

**Angular:** Es una plataforma de desarrollo, construida sobre Mecanografiado (Typescript). Como plataforma, Angular incluye ("angular.io," 2022):

- Un marco basado en componentes para crear aplicaciones web escalables.
- Una colección de bibliotecas bien integradas que cubren una amplia variedad de funciones, incluido el enrutamiento, la gestión de formularios, la comunicación cliente-servidor y más.
- Un conjunto de herramientas de desarrollo para ayudarlo a desarrollar, compilar, probar y actualizar su código.

**Electron:** Es un marco para crear aplicaciones de escritorio utilizando JavaScript, HTML y CSS. Al incorporar Chromium y Node.js en su binario, Electron le permite mantener una base de código JavaScript y crear aplicaciones multiplataforma que funcionan en Windows, macOS y Linux, sin necesidad de experiencia en desarrollo nativo ("electronjs.org," 2022).

#### 1.5.5 Lenguaje Unificado de Modelado (UML)

Es un lenguaje para la especificación, visualización, construcción y documentación de sistemas, no solo de software. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, pues ha sido diseñado para modelar cualquier tipo de soluciones informáticas, arquitectura o cualquier otra rama (González & de la Caridad, 2019).

#### 1.5.6 Herramienta de Modelado

**Visual Paradigm:** Es una aplicación comercializada por la empresa china Visual Paradigm Internacional, y ofrece varios módulos destinados al análisis y desarrollo de sistemas, con la posibilidad de integración de equipos con funcionalidades para el desarrollo ágil (de Oliveira Pequeno, da Silva, de Souza, & Machado, 2020).

## **1.6 Conclusiones del capítulo**

El marco teórico realizado en el capítulo, permitió identificar las principales herramientas y tecnológicas para la creación de una Aplicación de escritorio del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED, así como la metodología y aspectos para la investigación de soluciones existentes, que demostró la necesidad de crear una solución propia en nuestro país.

## CAPÍTULO 2: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

Este capítulo tiene como objetivo describir las actividades realizadas durante el proceso de análisis y diseño, para modelar y dar respuesta al problema planteado, además se marcan las pautas, objetivos y restricciones teniendo en cuenta los posibles riesgos a los que se puede enfrentar el desarrollo, lo cual permite crear una base sólida para una buena ejecución de las funcionalidades deseadas. Se realizan descripciones de los requisitos funcionales y no funcionales que se deben tener en cuenta para satisfacer las necesidades del cliente, las historias de usuario y se define la arquitectura y los patrones de diseño utilizados.

### 2.1 Modelado del dominio

De acuerdo con Craig Larman, un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés (Larman, 2004).

Utilizando la notación UML, un modelo del dominio se representa con un conjunto de diagramas de clases en los que no se define ninguna operación. Pueden mostrar (Larman, 2004):

- ✓ Objetos de dominio o clases conceptuales.
- ✓ Asociaciones de las clases conceptuales.
- ✓ Atributos de las clases conceptuales.

#### 2.1.1 Diagrama de proceso de negocio

La Gestión de Procesos de Negocios, en inglés Business Process Management, mejor conocida como BPM, es un enfoque sistemático de apoyo a la gestión empresarial que se concentra en cada uno de los procesos que conforman un negocio o empresa, con el objeto de mejorar su rendimiento. En otras palabras, BPM aporta una forma de administrar y mejorar el desempeño organizativo al optimizar y gestionar los procesos de negocio a través de un ciclo de modelado, ejecución y evaluación, combinando las mejores prácticas para orquestar las actividades de la organización con las personas en ellas involucradas, controlando la secuencia y gestionando a la vez las comunicaciones y los documentos asociados (Sandoval, 2017).

Los procesos de negocio son la secuencia de actividades relacionadas en un orden específico, con el fin de agregar valor a los productos o servicios de una organización. La modelación de estos procesos se lleva a cabo con la Notación para el Modelado de Procesos de Negocio o Business Process Model and Notation (BPMN), el cual es un estándar expuesto en el Grupo de Administración de Objetos o Object Management Group (OMG),

que define una notación gráfica para representar un modelo de procesos de negocio en un formato de flujo de trabajo (Sandoval, 2017).

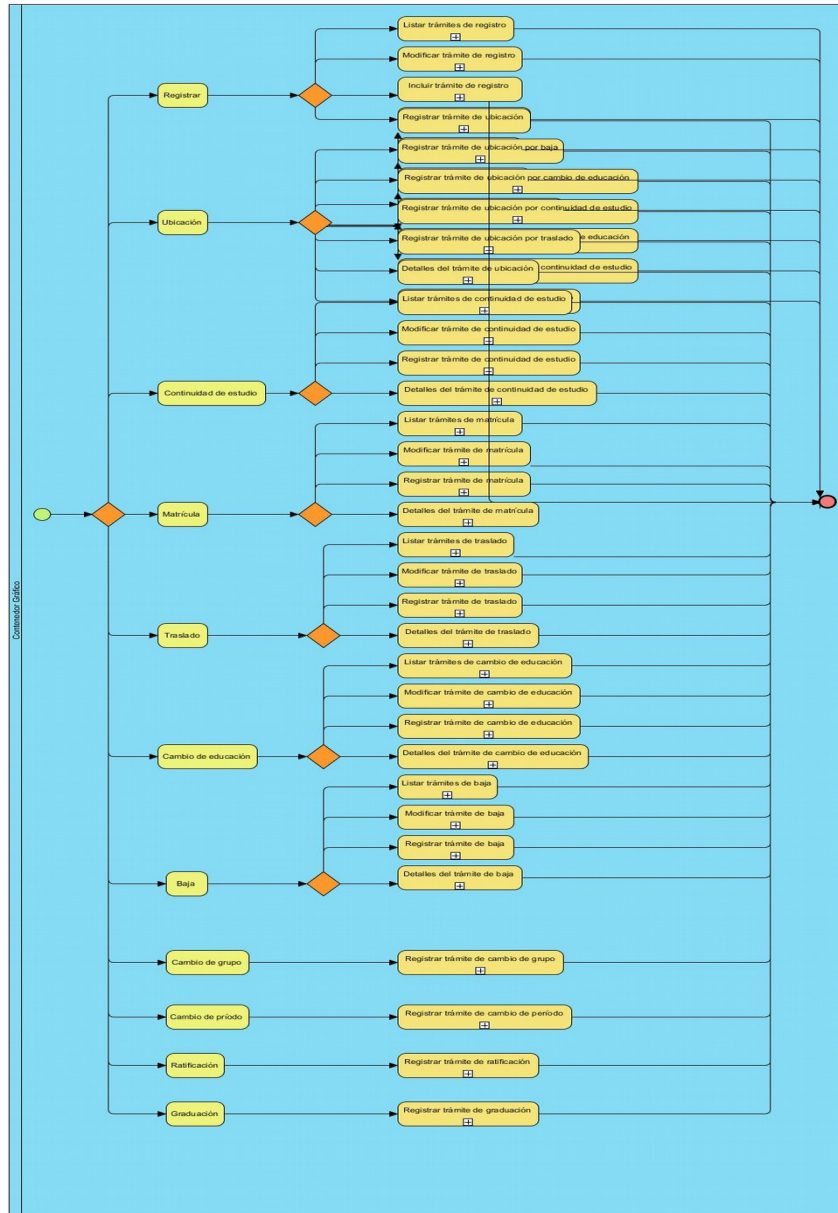


Figura 1: Diagrama del proceso de negocio "Ver datos de un trámite de registro"

### 2.1.2 Modelo Conceptual

Un modelo conceptual es todo aquello que nos permite conseguir una abstracción lógico-matemática del mundo real y que facilita la comprensión del problema a resolver (Sebastián Rubén Gómez Palomo, 2020).

2.1 Trámites Docentes

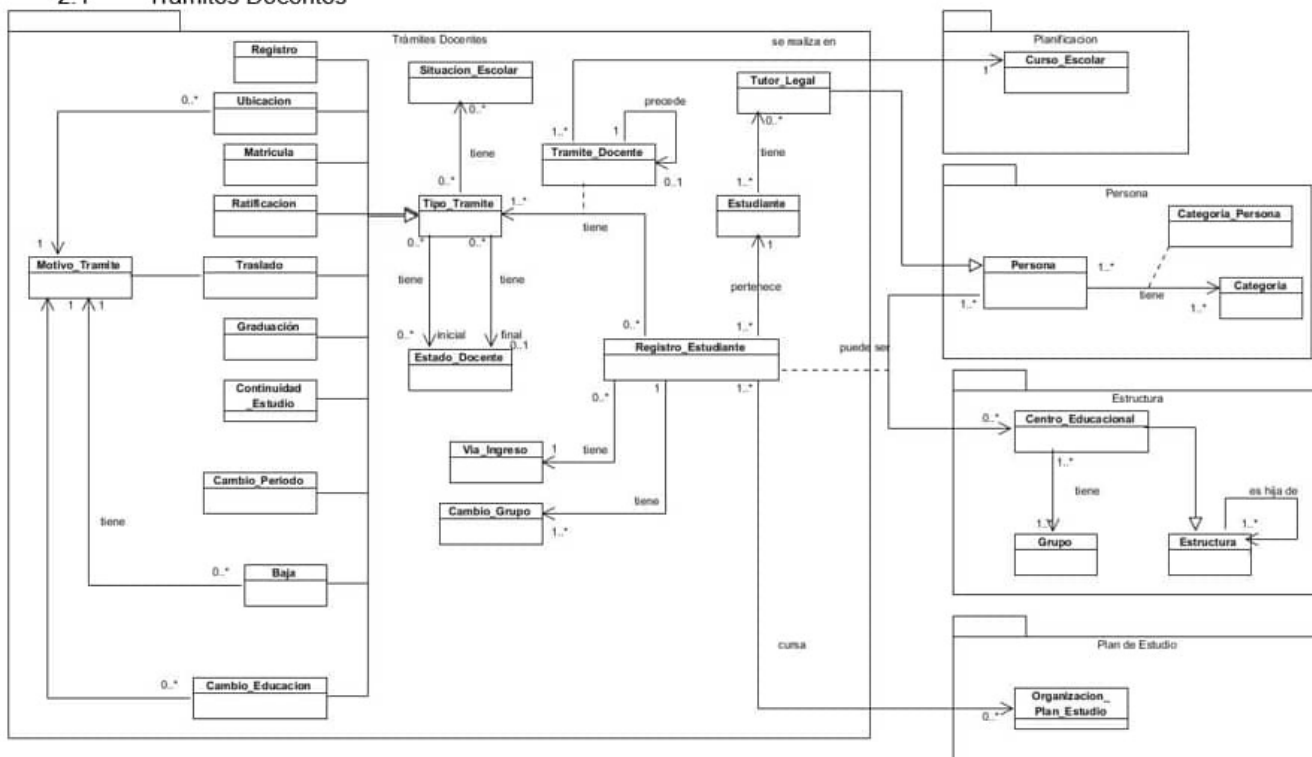


Figura 2: Modelo Conceptual

2.2 Requisitos del sistema

El esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto. Existen tres formas de encapsular los requisitos [Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por proceso (DRP)], agrupados en cuatro escenarios condicionados por el Modelado del negocio (Sánchez, 2015).

En cualquier proyecto de software los requisitos son las necesidades del producto que se debe desarrollar. Por ello, en la fase de análisis de requisitos se deben identificar claramente estas necesidades y documentarlas. Como resultado de esta fase se debe producir un documento de especificación de requisitos en el que se describa lo que el futuro sistema debe hacer. Por tanto, no se trata simplemente de una actividad de análisis, sino también de síntesis (Monferrer, 2001).

2.2.1 Requisitos Funcionales:

- (RF1) Incluir trámite de registro
- (RF2) Modificar trámite de registro

- (RF3) Listar trámites de registro
- (RF4) Ver datos de un trámite de registro
- (RF5) Listar trámites de ubicación
- (RF6) Ver datos de trámite de ubicación
- (RF7) Registrar trámite de ubicación
- (RF8) Modificar trámite de ubicación
- (RF9) Registrar trámite de ubicación por traslado
- (RF10) Modificar trámite de ubicación por traslado
- (RF11) Registrar trámite de ubicación por cambio
- (RF12) Modificar trámite de ubicación por cambio
- (RF13) Registrar trámite de ubicación por baja
- (RF14) Modificar trámite de ubicación por baja
- (RF15) Registrar trámite de matrícula
- (RF16) Modificar trámite de matrícula
- (RF17) Listar trámites de matrícula
- (RF18) Ver datos de un trámite de matrícula
- (RF19) Registrar trámite de traslado
- (RF20) Modificar trámite de traslado
- (RF21) Listar trámites de traslado
- (RF22) Ver datos de un trámite de traslado
- (RF23) Registrar trámite de cambio de educación
- (RF24) Modificar trámite de cambio de educación
- (RF25) Listar trámites de cambio de educación
- (RF26) Ver datos de un trámite de cambio de educación
- (RF27) Registrar trámite de cambio de grupo
- (RF28) Registrar trámite de ratificación de matrícula
- (RF29) Registrar trámite de cambio de período
- (RF30) Registrar trámite de graduación

**Tabla 3:** Especificación de los requisitos funcionales: Registrar trámite de registro, modificar trámite de registro, Listar trámite de registro y Ver datos de un trámite de registro.

No	Nombre	Descripción	Prioridad	Complejidad
RF1.	Incluir trámite de registro	El sistema debe permitir registrar un trámite de registro.	Alta	Media
RF2.	Modificar	El sistema debe permitir	Baja	Media

	trámite de registro	modificar un trámite de registro de una persona al cual no se le ha hecho una ubicación.		
RF3.	Listar trámites de registro	El sistema debe permitir listar todos los trámites de registro que correspondan a personas a las cuales no se les ha hecho una ubicación.	Media	Baja
RF4.	Ver datos de un trámite de registro	El sistema debe permitir consultar los datos de un trámite docente de registro.	Baja	Baja

### 2.2.2 Requisitos no funcionales:

#### Software:

(RFN1) Chrome Versión 105.0.5195.127 o Firefox 105.0.

(RFN2) Soporte de javascript en el navegador.

#### Hardware:

(RFN1) Computador con 4gb de memoria RAM o superior.

### 2.2.3 Descripción de requisitos por procesos

Tabla 4: Descripción textual del requisito “Ver datos de un trámite de registro”

<b>Precondiciones</b>	El usuario debe estar autenticado en el sistema. Debe existir registrado un trámite de registro en el sistema.
<b>Flujo de eventos</b>	
<b>Flujo básico Ver datos de un trámite de registro</b>	
1.	El usuario selecciona la opción <b>Icono Ver detalles</b> en las opciones internas del listado de trámites de registros.
1.	El sistema debe permitir ver los datos de un trámite de registro. Permite, además, seleccionar la siguiente opción: <ul style="list-style-type: none"> <li>• <b>Botón Cerrar:</b> Se cierra la IU y se pasa a la IU de precedencia correspondiente al listado de trámites de registro. Ver documento: FORTES_Levantamiento_AKADEMOS_MINED_DRP_Listar trámites de registro.odt</li> </ul>



2.	Concluye así el requisito.	
<b>Pos-condiciones</b>		
1.	Se mostraron los datos de un trámite de registro.	
<b>Validaciones</b>		
	FORTES_Levantamiento_AKADEMOS_MINED_Modelo_conceptual_Tramites_Docentes.odt	
	FORTES_Levantamiento_AKADEMOS_MINED_Modelo_conceptual_Persona.odt	
<b>Conceptos</b>	<b>Trámite_Docente</b>	Visibles en la interfaz: <ul style="list-style-type: none"> <li>- primer_nombre</li> <li>- segundo_nombre</li> <li>- primer_apellido</li> <li>- segundo_apellido</li> <li>- carne</li> <li>- sexo</li> <li>- foto</li> <li>- situacion_escolar</li> <li>- curso_escolar</li> <li>- tramitador</li> <li>- fecha_fin_tramite</li> <li>- estado_tramite</li> </ul> Utilizados internamente: NA
	<b>Persona</b>	Visibles en la interfaz: <ul style="list-style-type: none"> <li>- primer_nombre</li> <li>- segundo_nombre</li> <li>- primer_apellido</li> <li>- segundo_apellido</li> <li>- carne</li> <li>- dirección</li> <li>- municipio</li> <li>- provincia</li> <li>- consejo_popular</li> <li>- localidad</li> </ul> Utilizados internamente: <ul style="list-style-type: none"> <li>- NA</li> </ul>
	<b>Estudiante</b>	Visibles en la interfaz: <ul style="list-style-type: none"> <li>- hueroano</li> </ul> Utilizados internamente: <ul style="list-style-type: none"> <li>- NA</li> </ul>
	<b>Registro_Estudia nte</b>	Visibles en la interfaz: <ul style="list-style-type: none"> <li>- estado_docente</li> </ul> Utilizados internamente: <ul style="list-style-type: none"> <li>- NA</li> </ul>
	<b>Tutor_legal</b>	Visibles en la interfaz: <ul style="list-style-type: none"> <li>- parentesco</li> <li>- vinculación_laboral</li> <li>- procedencia_social</li> <li>- centro_trabajo</li> <li>- sector_laboral</li> </ul> Utilizados internamente: NA
<b>Requisitos especiales</b>	NA	
<b>Asuntos</b>	NA	

### 2.3 Diseño

La fase de diseño es técnicamente la parte central de la ingeniería del software durante la que se desarrollan, revisan y documentan los refinamientos progresivos de las estructuras de datos, de la estructura del programa y de los detalles procedimentales. El diseño es la única forma de traducir con precisión los requisitos del cliente en un producto o sistema acabado. Proporciona una imagen completa del software, desde una perspectiva de implementación, de forma que el sistema cumpla con parámetros como extensibilidad, reusabilidad, compatibilidad, portabilidad y robustez, lo que permite que todo el software alcance alto nivel de calidad. Es la base de todas las partes posteriores del desarrollo y de la fase de prueba (de diseño según Taylor).

#### 2.3.1 Modelo de Datos

Se puede definir como Modelo de Datos a las abstracciones mediante las cuales puede realizarse una representación de los problemas que se desea resolver. Es la estructura de una base de datos (Cabello, 2010).

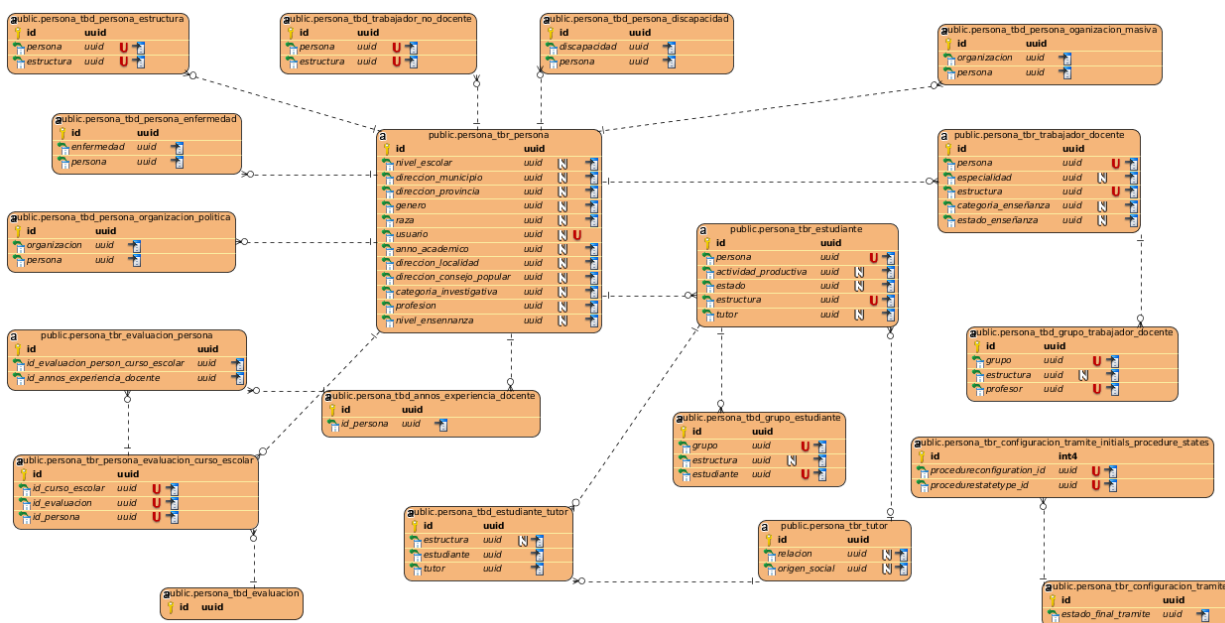


Figura 3: Modelo de Datos

Contiene los datos del estudiante, su relación con los centros educacionales y carreras. Además, todos los trámites asociados al mismo.

### 2.4 Arquitectura del Sistema

La arquitectura es un nivel de diseño que consiste en un conjunto de patrones y abstracciones coherente que proporcionan el marco de referencia necesario para guiar la construcción del software, cumpliendo los objetivos

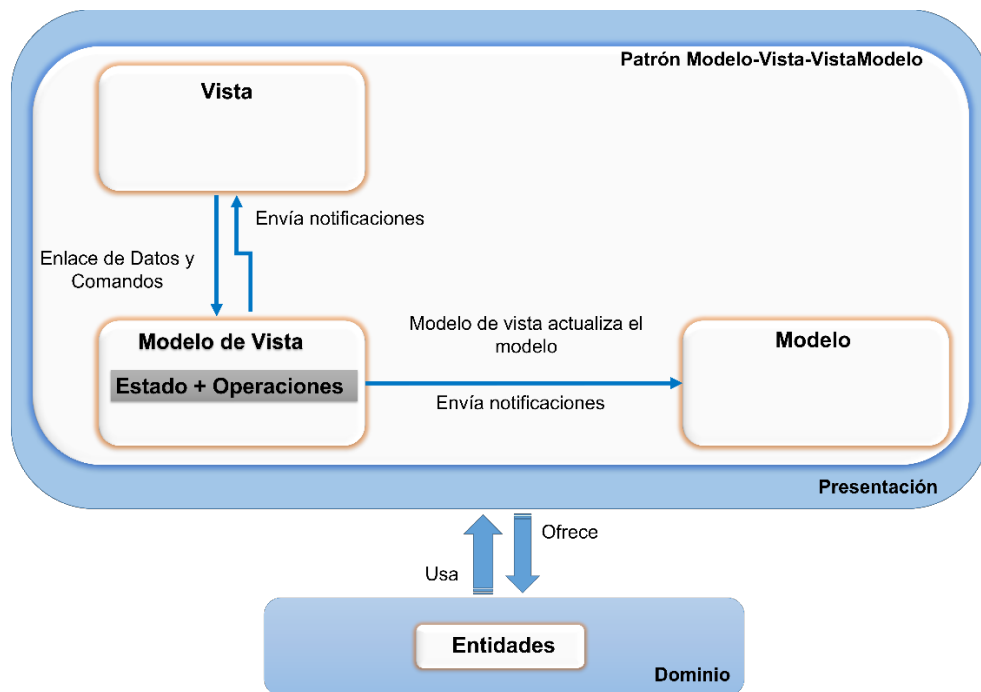
del sistema. Tiene como misión principal la identificación de requerimientos críticos en la estructura del sistema y la reducción de riesgos asociados a la construcción del mismo. Debe orientarse a las necesidades del cliente y ser flexible en cuanto a futuras modificaciones, tanto de software como de hardware, evitando que efectos inesperados puedan tener repercusión en el sistema (Garland, Shaw, & Engineering, 1993).

A partir del análisis realizado de la versión actual del Sistema de Gestión XAUCE AKADEMOS para el MINED y las características que presenta la herramienta a desarrollar, se utilizó el estilo arquitectural En Capas (N-capas).

El estilo arquitectural En Capas (N-capas) se basa en una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver. Los roles indican el tipo de forma de interacción con otras capas, las responsabilidades y la funcionalidad que implementan (De la Torre Llorente, Castro, Barros, & Nelson, 2010).

**Capa de presentación:** esta capa incluye diferentes elementos, como Vistas, Controladores, Modelos, entre otros que por lo general están estructurados siguiendo patrones de diseño propios de esta capa. Sus responsabilidades radican en presentar al usuario los conceptos de negocio mediante interfaces de usuarios, facilitar la explotación de dichos procesos, informar sobre el estado de los mismos y validar la entrada y salida de datos (De la Torre Llorente et al., 2010).

**Capa de dominio:** esta capa contiene componentes que implementan funcionalidades importantes del sistema, encapsulando toda su lógica para de esta forma apoyar los procesos que realizan los usuarios. Básicamente suelen ser clases que implementan la lógica del dominio dentro de sus métodos (De la Torre Llorente et al., 2010).



**Figura 4:** Arquitectura basada en el estilo arquitectural En Capas.

## 2.5 Patrones de diseño

En el marco de la arquitectura de software, el proceso de diseño de la arquitectura juega un papel muy importante ya que se tratan los temas más importantes a la hora de definir el sistema. Los patrones de diseño arquitecturales establecen una forma estándar de definir o implementar el aspecto del sistema que está siendo diseñado (De la Torre Llorente et al., 2010). A su vez existen otros patrones de diseño que proporcionan una estructura que resuelve un problema de diseño particular a través de una solución ya probada (Larman, Rodríguez, & Anaya, 1999).

### 2.5.1 Modelo-Vista-Modelo de Vista

MVVM es un patrón que permite cambiar las interfaces de usuario de una aplicación sin tener que alterar la lógica de la programación. Esto le aporta al software mayor flexibilidad para el desarrollo y mantenimiento del mismo. La esencia de MVVM es separar la Vista del Modelo introduciendo entre ellos una capa abstracta "Modelo Vista" (Sorensen & Mikhailesc, 2010).

EL Modelo representa los datos y la información del dominio, pero no los eventos ni las acciones que la manipulan. La Vista se encarga de mostrar la información de una forma clara y presentable. Una vista puede también contener ciertos comportamientos que eventualmente ejercen influencia en las propiedades del modelo (Amigo Portilla, 2015).

En MVVM la vista contiene comportamientos, eventos y enlaces a datos. Aunque tales eventos y comportamientos pueden ser asociados a propiedades, métodos y comandos, la vista es aún responsable de manejar sus propios eventos y no pasa esta tarea al modelo de vista (Sorensen & Mikhailesc, 2010).

El Modelo de vista mantiene al modelo separado y protegido de los detalles de la vista. Por eso es que el modelo de vista es la pieza clave del patrón. Esta separación permite que el modelo se limite a contener los datos y la vista solo a presentarlos. De esta forma el modelo de vista trabaja como intermediario entre ambos, recibiendo información de la vista e insertándola en el modelo y obteniendo datos del modelo para luego convertirlos en propiedades que pueden ser usadas por la vista. Todo este proceso es garantizado mediante métodos, comandos y otros puntos de acceso que ayudan a mantener el estado de la vista, manipular el modelo en la respuesta a acciones de la vista y disparar eventos en la misma (Amigo Portilla, 2015).

### 2.5.2 Patrones GRASP<sup>1</sup>

Para la realización del sistema se emplearán cinco patrones de la familia de los GRASP, estos son: Experto, Creador, Bajo Acoplamiento y Alta Cohesión. A continuación, se muestra la selección de los mismos:

---

<sup>1</sup> **GRASP:** Patrones generales para asignar responsabilidades o del inglés *General Responsibility Assignment Software Patterns*.

Empleando el patrón **Experto** se garantiza que cada clase del sistema asuma las responsabilidades que le conciernen, según las funcionalidades que se quieren implementar y a partir de la información que posee; por lo que cada clase contendrá la información necesaria para cumplir su responsabilidad.

**Bajo acoplamiento:** El patrón Bajo acoplamiento es una medida de la fuerza con que una clase se relaciona con otras, porque las conoce y recurre a ellas; una clase con bajo acoplamiento no depende de muchas otras. El uso de este patrón permite que las clases no se afecten por cambios de otros componentes, haciendo posible que sean fáciles de entender y de reutilizar (Cabeza Chávez, 2015).

**Alta Cohesión:** Se aplica para realizar un diseño que evite contener clases con un alto grado de abstracción, delegando las responsabilidades muy complejas a otros objetos. Este patrón se evidencia en las validaciones.

#### **Conclusiones del capítulo**

- ✓ La Obtención de los requisitos funcionales y no funcionales, a partir del proceso de identificación de los mismos, sirvió de guía para desarrollar las distintas funcionalidades, y de este modo satisfacer las necesidades del cliente.
- ✓ Los artefactos generados según la metodología de desarrollo utilizada y el empleo de los patrones descritos, constituyeron una guía fundamental para la construcción de la propuesta de solución.

## CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Este capítulo centra la atención en la descripción del proceso de construcción de la Aplicación de escritorio del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED, además de las pruebas realizadas al producto. Se exponen los estándares de codificación empleados y se establece la estrategia de pruebas para verificar la calidad del módulo implementado, así como la documentación referente a las pruebas seleccionadas. Además de la aplicación de la técnica IADOV para evaluar la propuesta de solución.

### 3.1 Estándar de codificación

**Indentación:** Emplear 4 espacios por cada nivel de indentación.

```
ngOnInit(): void {
    |   this.loadData();
    |   this.permissionService.loadPermissions(this.userPermissions.getPermissions());
    | }
}
```

*Figura 5: Indentación*

**Tabuladores y espacios:** No mezclar tabulaciones con espacios.

```
details($event) {
    |   if ($event.tn_parent_id != null) {
    |       |   this.loadingSpinnerService.show();
    |       |   this.structureService.getById($event.tn_parent_id).subscribe((data: any) => {
    |       |       |   this.parent = data.name;
    |       |       |   this.createModal($event);
    |       |       | });
    |       | }
    |   } else {
    |       |   this.parent = '';
    |       |   this.createModal($event);
    |   }
}
```

*Figura 6: Tabuladores y espacios.*

## Importaciones:

```
import { Component, OnInit } from '@angular/core';
import { NgxPermissionsService, NgxRolesService } from 'ngx-permissions';
import { FortesDataTable } from '../../../core/utils/FortesDataTable';
```

*Figura 7: Importaciones. Elaboración propia.*

**Declaraciones:** Se deben declarar cada variable en una línea diferente y de esta forma se puede comentar cada variable por separado.

```
//formulario
form: FormGroup;

//steps para manejar formulario en partes
stepOne: FormGroup;
stepTwo: FormGroup;
```

*Figura 8: Declaraciones de variables.*

**Tamaño máximo de líneas:** Limita todas las líneas a un máximo de 79 caracteres.

```
tn_parent_id: [null, [Validators.required]],
address: [null,
  [
    Validators.required,
    Validators.maxLength(100)
  ]
],
```

*Figura 9: Tamaño máximo de líneas.*

**Comentarios:** Los comentarios deben añadir claridad al código, un comentario que contradiga el código es peor que no tener ningún comentario.

```
//navega al listado de escuelas
confirmCancel() {
  this.router.navigate(['/structures/school']).then(() => {});
}
```

*Figura 10: Comentarios.*

**Asignación de nombres:** Evitar nombres largos y que difieran en una letra o en el uso de mayúsculas. Para nombrar funciones y variables se describe con la primera palabra en

minúscula en caso de ser compuesta se utiliza la notación *CamelCase*<sup>2</sup>.

Para nombres simples:

```
update() {  
  const isdisable = this.formUse.get('is_disable').value;  
  this.formUse.get('is_disable').setValue(!isdisable);  
  this.loadingSpinnerService.show();  
  this.manager.update(this.ID, this.formUse.value).subscribe(  
    () => {  
      this.message.success(this.applicationMessages.actionsMessages.succes);  
      this.router.navigate(['/structures/school']).then(() => {  
        this.loadingSpinnerService.hide();  
      });  
    },  
    (error) => {  
      this.loadingSpinnerService.hide();  
      this.message.error(this.applicationMessages.systemMessages.errorAPiConnection);  
    }  
  );  
}
```

**Figura 11:** Asignación de nombres simples.

Para nombres compuestos:

```
changeProvince($event) {  
  if ($event !== null) {  
    this.province.getById($event).subscribe((data: any) => {  
      this.provinceName = data.name;  
      this.codeProvince = data.code;  
    });  
  } else {  
    this.provinceName = '';  
    this.codeProvince = '';  
  }  
}
```

**Figura 12:** Asignación de nombres compuestos.

**Líneas en blanco:** Las definiciones de métodos dentro de una clase son separados por una línea en blanco.

---

<sup>2</sup> *CamelCase* es la práctica de escribir frases o palabras compuestas eliminando los espacios y poniendo en mayúscula la primera letra de cada palabra.



```

changeMunicipality($event) {
  if ($event !== null) {
    this.municipality.getById($event).subscribe((data: any) => {
      this.codeMunicipality = data.code;
    });
  } else {
    this.codeMunicipality = '';
  }
}
}

//navega al listado de escuelas
confirmCancel() {
  this.router.navigate(['/structures/school']).then(() => {});
}
}

```

Figura 13: Líneas en blanco.

### 3.1.2 Diagrama de componentes

Este tipo de diagrama representa cómo un sistema de *software* es dividido en componentes, los cuáles pueden ser archivos, bibliotecas, módulos, paquetes o ejecutables y muestra el comportamiento de los servicios que estos componentes proporcionan y utilizan a través de las interfaces. A continuación, se muestra el diagrama de componentes del sistema (Portales López, 2016):

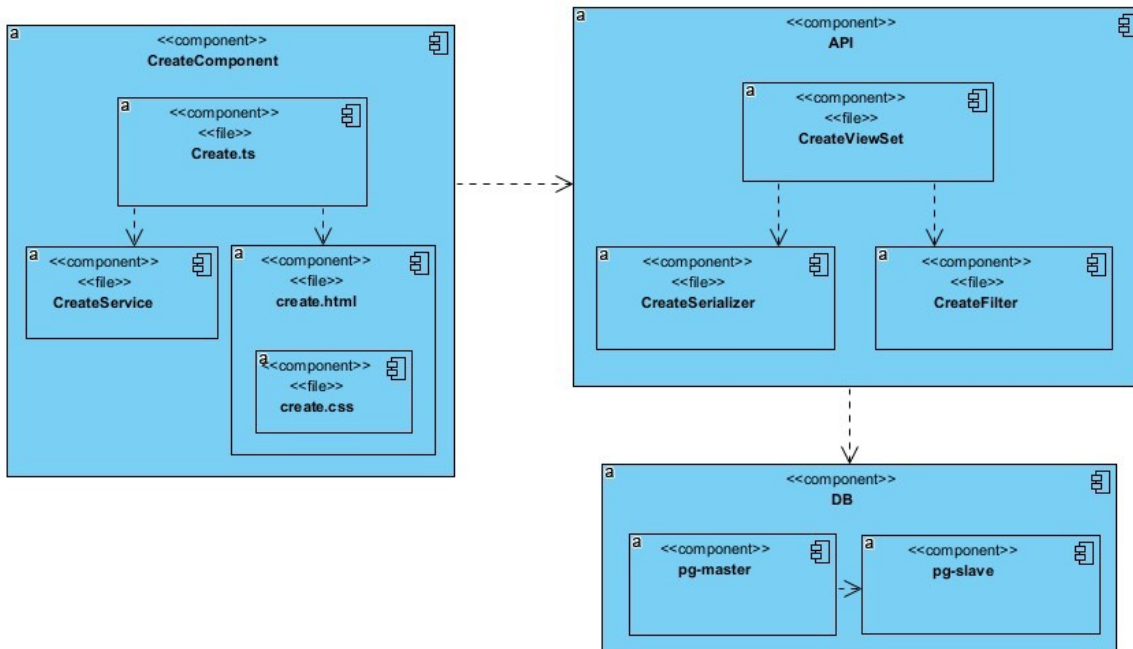


Figura 14: Diagrama de Componentes de la funcionalidad genérica Crear.

### 3.1.2 Diagrama de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Se utiliza tanto en el diseño como en la implementación y está compuesto por nodos, componentes que residen en los nodos y las relaciones entre estos. Un nodo es un elemento físico que representa un recurso computacional utilizado para modelar la topología del hardware sobre el que se ejecuta el sistema. Estos son conectados por asociaciones de comunicación, tales como enlaces de red, conexiones TCP/IP, entre otros (Pons Ramírez, Pérez Rodríguez, González Mesa, & González Yera, 2016).

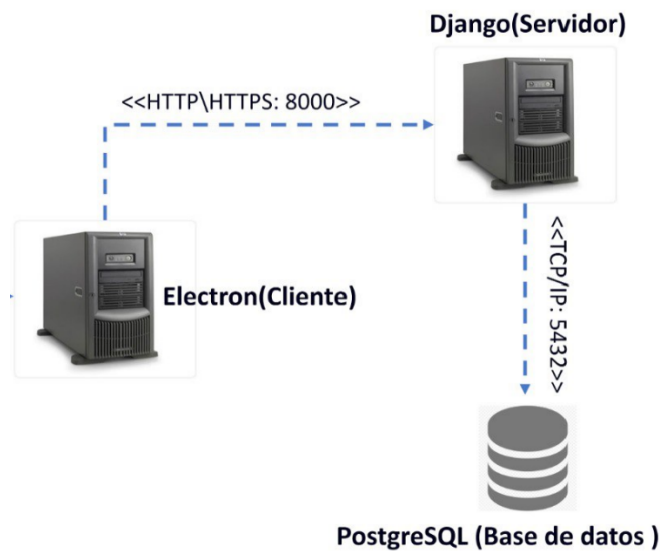


Figura 15: Diagrama de despliegue.

### 3.3 Pruebas de software

Las pruebas de software son un conjunto de herramientas, técnicas y métodos que evalúan la excelencia y el desempeño de un software, involucra las operaciones del sistema bajo condiciones controladas y evaluando los resultados (Pressman, 2005).

#### 3.3.1 Pruebas a realizar

##### Funcionales

Se escoge este nivel de prueba ya que lo que se desea es la verificación de las funcionalidades del sistema a través de las interfaces y su relación con los demás

componentes del sistema.

Las pruebas de sistema tienen como objetivo ejercitar profundamente el sistema comprobando la integración del sistema de información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de sistemas de información con los que se comunica (Pressman, 2005). En lugar de caer en este absurdo, el ingeniero del software debe de anticiparse a posibles problemas de la interfaz (Cabeza Chávez, 2015):

- ✓ Diseñar ruta de manejo de errores que prueben toda la información proveniente de otros elementos del sistema.
- ✓ Aplicar una serie de pruebas que simulen datos incorrectos u otros posibles errores en la interfaz de software.
- ✓ Registrar los resultados de la prueba como evidencia en el caso que se culpe.

### **Unitarias**

La prueba de unidad se concentra en el esfuerzo de verificación de la unidad más pequeña del diseño del software: el componente o módulo de software. Tomando como guía la descripción del diseño a nivel de componentes, se prueban importantes caminos de control para descubrir errores dentro de los límites del módulo. Las pruebas de unidad se concentran en la lógica del procesamiento interno y en las estructuras de datos dentro de los límites de un componente (Pressman, 2005).

### **Integración**

La prueba de integración es una técnica simétrica para construir la arquitectura del software mientras, al mismo tiempo, se aplican las pruebas para descubrir errores asociados con la interfaz. El objetivo de tomar componentes a los que se aplicó una prueba de unidad y construir una estructura de programa que determine el diseño (Pressman, 2005). La estrategia para realizar la prueba de integración seleccionada por el autor de la investigación científica es la integración ascendente.

La estrategia de integración ascendente como su nombre lo indica, empieza la construcción y la prueba con componentes de los niveles más bajos de la estructura del programa (Pressman, 2005).

### 3.3.2 Método de prueba

#### Caja Negra

Se escoge el método de caja negra ya que los niveles de pruebas escogidos pretenden probar el funcionamiento de la interfaz y de las funcionalidades del sistema, para esto es necesario este método ya que permite comprobar el comportamiento del software a través de los requisitos funcionales (Pressman, 2005), obviando el funcionamiento interno y la estructura del programa.

Las pruebas de caja negra pretenden encontrar estos tipos de errores:

- Funciones incorrectas o ausentes.
- Errores en la interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de comportamiento o desempeño.
- Errores de inicialización y de terminación.



*Figura 16: Esquema del Método de Caja Negra.*

#### Caja Blanca

La prueba de caja blanca, en ocasiones llamada prueba de caja de cristal, es un método de diseño que usa la estructura de control descrita como parte del diseño al nivel de componentes para derivar los casos de prueba. Al emplear los métodos de prueba de caja blanca, el ingeniero del software podrá derivar casos de prueba que (Pressman, 2005):

1. Garanticen que todas las rutas independientes dentro del módulo se han ejercitado por lo menos una vez.
2. Ejerciten los lados verdadero y falso de todas las decisiones lógicas.

3. Ejecuten todos los bucles en sus límites y dentro de sus límites operacionales.
4. Ejerciten estructuras de datos internos para asegurar su validez.

### **3.3.3 Técnica de prueba**

#### **Partición equivalente**

##### **Características**

- Divide el dominio de entrada de un programa en clases de datos, a partir de las cuales pueden derivarse casos de prueba.
- Descubre clases de errores, que, de otra manera, requeriría la ejecución de muchos casos antes de que se observe el error general.
- Reducir al máximo el total de casos de prueba que deben desarrollarse (Cabeza Chávez, 2015).

##### **El diseño consiste:**

- Identificar clases de equivalencia. Las clases de equivalencia son un conjunto de estados válidos o inválidos para condiciones de entrada.
- Crear los casos de prueba.

##### **Camino básico**

La prueba de camino básico es una técnica de prueba de caja blanca. Este método permite que el diseñador de casos de pruebas obtenga una medida de complejidad lógica de un diseño procedimental y que use esta medida como guía para definir un conjunto básico de rutas de ejecución. Los casos de prueba derivados para ejercitar el conjunto básico deben garantizar que se ejecutan cada instrucción del programa por lo menos una vez durante la prueba (Pressman, 2005).

### **3.4 Aplicación de las pruebas**

#### **3.4.1 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 (Sánchez, 2015).

### Pruebas unitarias

Las pruebas unitarias fueron realizadas mediante el método de caja blanca mediante la técnica camino básico. Las unidades de prueba más pequeña son las operaciones dentro de la clase. A continuación, se realiza la técnica del camino básico, al método `configureCorosync ()` en la clase `Main` en el paquete `backend`.

```
1 isElectron() {
2 // Renderer process
3 if (typeof window !== 'undefined' && typeof window.process === 'object' && window.process.type === 'renderer') {
4 // Main process
5 if (typeof process !== 'undefined' && typeof process.versions === 'object' && !process.versions.electron) {
6 // Detect the user agent when the `nodeIntegration` option is set to true
7 if (typeof navigator === 'object' && typeof navigator.userAgent === 'string' && navigator.userAgent.indexOf('Electron') >= 0) {
8 return false;
9 }
```

Figura 17: Enumeración de las líneas de código.

Luego de enumerar las líneas de código, se diseña la gráfica del programa que describe el flujo de control lógico empleando nodos y aristas.

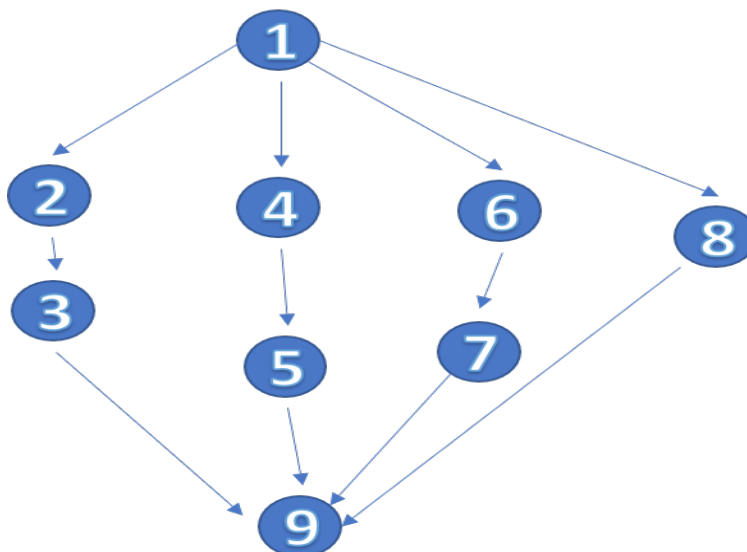


Figura 18: Grafo de flujo.

A partir del grafo obtenido con 9 nodos y 11 aristas, se calcula la complejidad dicromática  $V(G)$  la cual constituye una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa.

$$V(G) = \text{cantidad\_aristas} - \text{cantidad\_nodos} + 2$$

$$V(G) = 11 - 9 + 2 = 4$$

Un camino independiente es cualquier camino del programa que introduce al menos un nuevo conjunto de sentencias de proceso o una nueva condición (Pressman, 2009). La cantidad de caminos independientes se establece por la complejidad ciclomática, por tanto, se identifican 4 caminos como se muestra en la siguiente tabla.

**Tabla 5:** Listado de caminos independientes.

No.	Camino
1	1-2-3-9
2	1-4-5-9
3	1-6-7-9
4	1-8-9

El valor de la complejidad ciclomática ofrece además un límite superior para la cantidad de pruebas que se deben diseñar y ejecutar para garantizar que se cumplen todas las sentencias del programa (Pressman, 2005). A continuación, se muestran los casos de pruebas para cada camino independiente.

**Tabla 6:** Caso de prueba de unidad para el camino 1.

<b>Caso de prueba de unidad</b>	
<b>No. Camino:</b> 1	<b>Camino:</b> 1-2-3-9
<b>Nombre de la persona que realiza la prueba:</b> Yesenia Alfonso Ramirez	
<b>Descripción de la prueba:</b> Determinar si el entorno es electrón.	
<b>Entrada:</b> No	

<b>Resultado esperado:</b> Retorna verdadero.
<b>Evaluación de la prueba:</b> Satisfactoria. El valor retornado es verdadero.

*Tabla 7: Caso de prueba de unidad para el camino 2.*

<b>Caso de prueba de unidad</b>	
<b>No. Camino: 2</b>	<b>Camino: 1-4-5-9</b>
<b>Nombre de la persona que realiza la prueba:</b> Yesenia Alfonso Ramirez	
<b>Descripción de la prueba:</b> Determinar si el entorno es electrón.	
<b>Entrada:</b> No	
<b>Resultado esperado:</b> Retorna verdadero.	
<b>Evaluación de la prueba:</b> Satisfactoria. El valor retornado es verdadero.	

*Tabla 8: Caso de prueba de unidad para el camino 3.*

<b>Caso de prueba de unidad</b>	
<b>No. Camino: 3</b>	<b>Camino: 1-6-7-9</b>
<b>Nombre de la persona que realiza la prueba:</b> Yesenia Alfonso Ramirez	
<b>Descripción de la prueba:</b> Determinar si el entorno es electrón.	
<b>Entrada:</b> No.	
<b>Resultado esperado:</b> Retorna verdadero.	
<b>Evaluación de la prueba:</b> Satisfactoria. El valor retornado es verdadero.	

<b>Caso de prueba de unidad</b>	
<b>No. Camino: 4</b>	<b>Camino: 1-8-9</b>
<b>Nombre de la persona que realiza la prueba:</b> Yesenia Alfonso Ramirez	
<b>Descripción de la prueba:</b> Determinar si el entorno es electrón.	
<b>Entrada:</b> No.	
<b>Resultado esperado:</b> Retorna falso.	
<b>Evaluación de la prueba:</b> Satisfactoria. El valor retornado es falso.	



## Resultado de las pruebas unitarias

Fueron realizadas tres iteraciones mediante el método de caja blanca mediante la técnica de camino básico donde fue encontrado un error de validación para la escritura en fichero y un error de llamada a métodos.

## Pruebas Funcionales

Las pruebas funcionales fueron realizadas mediante en el método de caja negra mediante la técnica partición equivalente.

## Caso de Prueba 2. Guard para seguridad en electrón.

### Descripción de las variables

**Tabla 9:** Descripción de las variables del caso de prueba "Es electron".

N o	Nombre de Campo	Clasificació n	Valor Nulo	Descripción
1	Es electron.	Boolean	no	Valor retornado por el método isElectron.

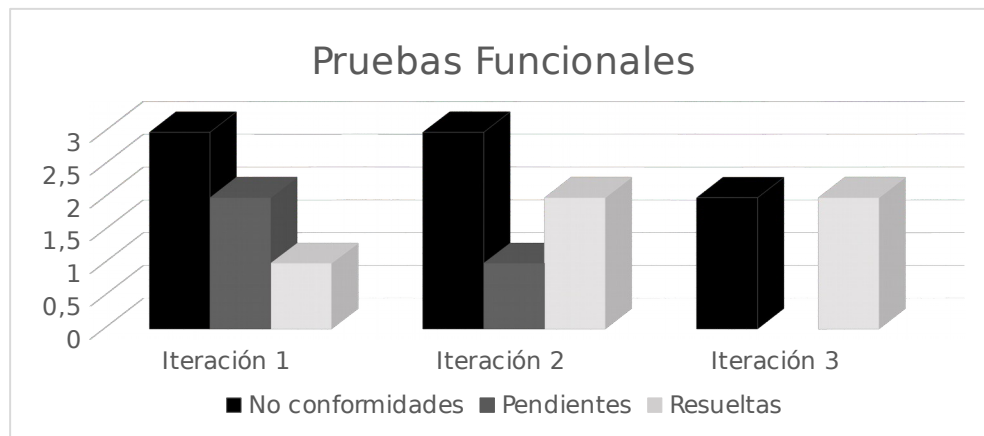
**Tabla 10:** Descripción de los escenarios del caso de prueba " Configuración del guard".

Escenario	Descripción	V1 IsElectron	Respuesta del sistema	Flujo central
EC 1.1 Configuración del guard con valor verdadero.	El valor retornado por el método isElectron es verdadero.	V true	El sistema muestra navega a la ruta seleccionada por el usuario.	1. El usuario selecciona una opción del menú de navegación.

EC	1.2	El valor retornado por el método isElectron es falso.	V falso	El sistema muestra una pantalla de acceso denegado y la opción para regresar al inicio.	
EC	1.3	El valor retornado por el método isElectron es incorrecto.	I asd	El sistema muestra una pantalla de acceso denegado y la opción para regresar al inicio.	

### Resultados de los casos de pruebas

Se utilizó el método de caja negra para realizar las pruebas sobre la interfaz de la Aplicación de escritorio del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED. Donde fueron encontradas 8 no conformidades en 3 iteraciones. De estas no conformidades 3 fueron de errores ortográficos y 5 de validación incorrecta. En la Figura 16 se muestran las iteraciones realizadas y las no conformidades detectadas, las resueltas y las pendientes.



*Figura 19: Estadísticas de las pruebas.*

### 3.4.2 Pruebas de Aceptación

Se escoge realizar la prueba de aceptación ya que es necesario para la validación de la solución que el cliente este de acuerdo con el funcionamiento del sistema desarrollado y así pueda emitir la carta de aceptación donde se demuestra la conformidad del cliente con la solución.

Como técnica se escoge las pruebas alfas esta se lleva a cabo por un cliente en el lugar de desarrollo. Estas pruebas se llevan a cabo en un entorno controlado y se usa el software de forma natural con el desarrollador como observador del usuario.

Para que tengan validez se debe crear un ambiente con las mismas condiciones que se encontrarán en las instalaciones del cliente. Una vez logrado esto se procede a realizar las pruebas y a documentar los resultados. El resultado de las pruebas de aceptación se muestra en el Anexo 2, donde fue emitida el acta de aceptación con el cliente.

### 3.5 Evaluación del objetivo general de la investigación

La técnica de ladov es utilizada para determinar el nivel de satisfacción individual y grupal de los usuarios a partir de una encuesta elaborada según las exigencias pertinentes. En la presente investigación fue aplicada a cinco especialistas, objeto de estudio. La encuesta realizada se muestra en el Anexo 3.

La técnica de ladov constituye una vía indirecta para el estudio de satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre las tres

preguntas cerradas, que se intercalan dentro de un cuestionario y cuya relación el encuestado desconoce. En este caso, de la encuesta, son las preguntas 2,3 y 4 se relacionan a través de lo que se denomina el “Cuadro Lógico de ladov” que se muestra en la tabla 11.

**Tabla 11:** Cuadro lógico de ladov.

4. Luego de haber mostrado los resultados de la solución refleje en qué medida le gusta la solución desarrollada.	2. ¿Considera usted correcta la forma en que se brindan los servicios actualmente a los usuarios?								
	<b>No</b>			<b>No sé</b>			<b>Sí</b>		
	3. ¿Considera usted factible el desarrollo de una aplicación de escritorio sin conexión a internet que permita el uso del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED?								
	<b>Sí</b>	<b>No sé</b>	<b>No</b>	<b>Sí</b>	<b>No sé</b>	<b>No</b>	<b>Sí</b>	<b>No sé</b>	<b>No</b>
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	3	4
No me gusta nada	6	6	6	6	4	4	6	6	5
No sé decir	2	3	6	3	3	3	6	6	4

La forma de utilizar la tabla es la siguiente: Cada encuestado recibe una evaluación individual en dependencia de las respuestas que dé a las preguntas cerradas. Para facilitar el procesamiento posterior, en el diseño de la encuesta se debe tener en cuenta que a estas preguntas sólo se responda de la forma prevista en el cuadro lógico de ladov. Las respuestas a las preguntas 2 y 3 pueden ser Sí, NO, NO SÉ, y a las preguntas 4, “Me gusta mucho”, “Me gusta más de lo que me disgusta”, “Me da lo mismo”, “Me disgusta más de lo que me gusta”, “No me gusta nada”, o “No sé qué decir”.

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y -1. El número resultante de la interrelación de las tres preguntas indica la posición de cada encuestado en

la siguiente escala de satisfacción: clara satisfacción +1, más satisfecho que insatisfecho 0.5, no definido y contradictorio 0, más insatisfecho que satisfecho -0.5 y clara insatisfacción -1

El índice de satisfacción grupal (ISG) se expresa en una escala numérica que va desde 1 (máxima satisfacción), hasta -1 (máxima insatisfacción). El ISG se calcula mediante la siguiente fórmula:

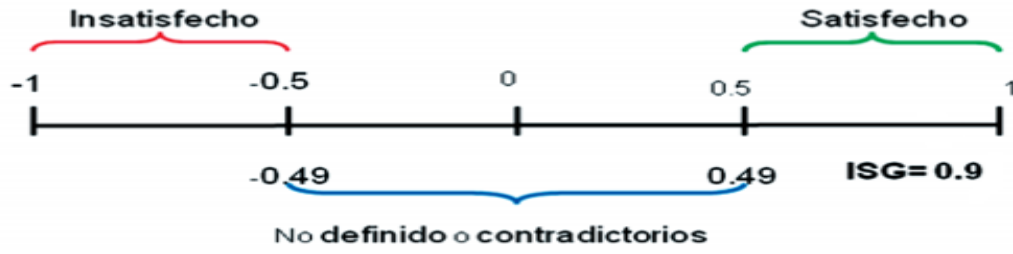
$$ISG = \frac{A(+1) + B(+0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

En esta fórmula A, B, C, D, E, representan la cantidad de encuestados colocados respectivamente en las posiciones de satisfacción 1; 2; 3 o 6; 4; 5 y donde N representa la cantidad total de encuestados

Por ejemplo, si 4 especialistas presentan máxima satisfacción, 1 están más satisfechos que insatisfechos, el índice de satisfacción se calcularía de la siguiente forma:

$$ISG = \frac{4(+1) + 1(+0.5) + 0(0) + 0(-0.5) + 0(-1)}{5}$$

**ISG=0.9**



*Figura 20: Ubicación del índice de satisfacción Grupal.*

Los valores de ISG que se encuentran comprendidos entre -1 y -0,5 indican insatisfacción; los comprendidos entre -0,49 y +0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción.

El proceso de evaluación del objetivo de la investigación mediante la técnica de Iadov confirmó su factibilidad de uso, expresando cuantitativamente en el alto ISG (0.9) y

cualitativamente en los criterios emitidos en el centro FORTES, lo que refleja la aceptación de la propuesta y el reconocimiento a su utilidad.

### **Conclusiones parciales**

- ✓ El uso de los diagramas de componentes y el de despliegue, permitió observar la estructura general del sistema y de la topología de *hardware* donde se ejecuta el mismo.
- ✓ La utilización de los estándares de codificación permitió obtener una estructura homogénea con menor cantidad de errores y de fácil mantenimiento.
- ✓ Se validó la propuesta de solución, luego de tres iteraciones de pruebas donde fueron corregidas las no conformidades detectadas.
- ✓ La prueba de aceptación realizada permitió evaluar la propuesta de solución por parte del cliente y su satisfacción con el producto final.
- ✓ La aplicación de la técnica de ladov permitió evaluar de manera satisfactoria la propuesta de solución demostrando la satisfacción de los potenciales usuario hacia el sistema desarrollado.

## **CONCLUSIONES FINALES**

- ✓ Al completar el desarrollo de la aplicación de escritorio se le dió cumplimiento al objetivo planteado inicialmente, obteniendo como resultado principal que el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED pueda ser usado sin conexión a internet.
- ✓ El uso de un conjunto de patrones y buenas prácticas de programación, permitió diseñar e implementar diferentes clases que dan cumplimiento a los requisitos funcionales y no funcionales de la propuesta de solución.
- ✓ Se logró obtener una propuesta que satisface los requisitos funcionales y no funcionales mediante las pruebas realizadas.

## **RECOMENDACIONES**

En aras de enriquecer la solución propuesta se propone la siguiente recomendación:

- ✓ Incorporar mecanismos para actualizar de manera automática la aplicación dentro de electrón.



## REFERENCIAS BIBLIOGRÁFICAS

- Ackoff, R. L. J. S. D. R. (1994). Systems thinking and thinking systems. *10*(2-3), 175-188.
- aGora. (2020). Retrieved from <https://www.agora-erp.com/es/editions/agoradesktop>
- Alkhars, A., & Mahmoud, W. (2017). Cross-Platform Desktop Development (JavaFX vs. Electron). In.
- Amigo Portilla, A. (2015). *Herramienta para la gestión de informes diagnósticos de los estudios imagenológicos realizados en el servicio de Hemodinámica*. Universidad de las Ciencias Informáticas. Facultad 2,
- angular.io. (2022). Retrieved from <https://angular.io/guide/what-is-angular#angular-applications-the-essentials>
- Angulo Luna, F. D. (2021). Sistema web para la optimización del proceso de gestión académica de la IEP Niño Jesús de Belén.
- Cabello, M. V. N. (2010). *Introducción a las bases de datos relacionales*: Vision Libros.
- Cabeza Chávez, Y. (2015). *Módulo de JAVA para la herramienta Auditoría de Código Fuente*. Retrieved from La Habana:
- Caldas, U. D. F. J. (Producer). (2021, 12 10). Sistema Gestión Académica. *Sistema Gestión Académica*. Retrieved from <https://www.udistrital.edu.co/sistema-gestion-academica>
- Chacon, J. S. V. (Producer). (2019, 06 10). Sistema de Gestion Academica (SGA). Retrieved from <https://sourceforge.net/projects/sistema-de-gestion-academica/>
- Code.visualstudio.com. (2022). Retrieved from <https://code.visualstudio.com/docs>
- ComputerWorld (Producer). (2003). Historia de los sistemas operativos. Retrieved from <https://www.computerworld.es/archive/historia-de-los-sistemas-operativos>
- de diseño según Taylor, D. Fundamentos del diseño de software.
- De la Torre Llorente, C., Castro, U. Z., Barros, M. A. R., & Nelson, J. C. (2010). Guía de Arquitectura N-Capas orientada al Dominio con .NET. In: España: Microsoft Iberica.
- de Oliveira Pequeno, P. V., da Silva, E. S. R., de Souza, J. B., & Machado, M. C. J. B. J. o. D. (2020). ConceptER-Uma ferramenta para criação e manutenção do Modelo Entidade-Relacionamento e geração automática de instruções SQL para banco de dados. *6*(7), 49345-49354.
- Developer.mozilla.org. (2015). Retrieved from [https://developer.mozilla.org/es/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics)
- Dori, D., Sillitto, H., Griego, R. M., McKinney, D., Arnold, E. P., Godfrey, P., . . . Krob, D. J. I. S. J. (2019). System definition, system worldviews, and systemness characteristics. *14*(2), 1538-1548.
- electronjs.org. (2022). Retrieved from <https://www.electronjs.org/docs/latest>
- EUROINNOVA (International Online Education). (2019). Retrieved from <https://www.euroinnova.edu.es/blog/nuevas-tecnologias-informaticas>
- Figuroa, R. G., Solís, C. J., & Cabrera, A. A. J. U. T. P. d. L., Escuela de Ciencias de la Computación. (2008). Metodologías tradicionales vs. metodologías ágiles. *9*(1), 1-10.
- Formación en Gestión Académica. (2018, 09 26). *Comisión Nacional de Acreditación CNA-Chile*. Retrieved from <https://www.cnachile.cl/noticias/Paginas/GESTI%C3%93N-ACAD%C3%89MICA.aspx>
- Garland, D., Shaw, M. J. A. i. S. E., & Engineering, K. (1993). An introduction to software architecture. *1*.

- Gonzales Chavez, L. F., & Andrade Vela, O. D. (2022). INFLUENCIA DE LAS HERRAMIENTAS TECNOLÓGICAS DE INFORMACIÓN EN LA GESTIÓN ADMINISTRATIVA DE LA OFICINA DE LOGÍSTICA DE LA DIRECCIÓN REGIONAL DE SALUD DE LORETO-IQUITOS-2022.
- González, H., & de la Caridad, D. (2019). *Sistema para la gestión de la guardia obrera en la Facultad 1 de la Universidad de las Ciencias Informáticas*. Universidad de las Ciencias Informáticas. Facultad 1.,
- Gutierrez, M. E. D. (2020). Universidad Técnica de Babahoyo. <http://dspace.utb.edu.ec/handle/49000/7643>
- Innovasoft (Producer). (2022). Sistema de Gestión Académica (CLASS). Retrieved from <https://www.innovasof.com/>
- Instituto de Energías Renovables de la UNAM. (2007, 05 04). *El concepto de Sistema*. Retrieved from <https://www.ier.unam.mx/~ojs/pub/Termodinamica/node9.html>
- Johnson, L. J. S. W. (2021). What is a System?
- Koskela, J. (2022). Serverless backend migration from JavaScript to TypeScript.
- Larman, C. (2004). *UML y Patrones*: Prentice Hall.
- Larman, C., Rodríguez, L. M. H., & Anaya, H. C. (1999). *UML y Patrones: Introducción al análisis y diseño orientado a objetos* (Vol. 2): Prentice Hall.
- México, U. I. C. (Producer). (2022). Sistema de Gestión Académica. *Sistema de Gestión Académica*. Retrieved from <https://enlinea.uia.mx/gestionacademica/login.cfm>
- Monferrer, A. R. (2001). *Especificación de Requisitos Software según el estándar de IEEE 830*. Retrieved from
- Pedamkar, P. (Producer). (2019). EDUCBA. Retrieved from <https://www.educba.com/what-is-desktop-software/>
- Pons Ramírez, A., Pérez Rodríguez, O., González Mesa, Y., & González Yera, Y. (2016). *Desarrollo del módulo de Simulaciones de osteotomía para el sistema de Planificación Quirúrgica Ortopédica*.
- Portales López, R. (2016). *Sistema para la selección de modelos de negocio en la comercialización de los productos del Centro de Informática Industrial*. Universidad de las Ciencias Informáticas. Facultad-5.,
- Pressman, R. S. (2005). *Software engineering: a practitioner's approach*: Palgrave macmillan.
- Sánchez, T. R. (2015). *Metodología de desarrollo para la Actividad productiva de la UCI*. Retrieved from La Habana:
- Sandoval, F. (2017). Gestión de Procesos de Negocios. *Revista del Centro de Investigación en Sistemas de Información*.
- Sebastián Rubén Gómez Palomo, E. A. M. G. (2020). *Aproximación a la Ingeniería del Software*: Editorial Universitaria Ramón Areces.
- Sorensen, E., & Mikailesc, M. J. M. J. (2010). Model-view-ViewModel (MVVM) design pattern using Windows Presentation Foundation (WPF) technology. 9(4), 1-19.
- Stallings, W., Aguilar, L. J., Doderó, J. M., Torres, E., & Mora, M. K. (1997). *Sistemas operativos* (Vol. 732): Prentice Hall.
- TechTerms. (2007). Retrieved from <https://techterms.com/definition/css>
- typescriptlang-org. (2022, Julio 13). *typescriptlang-org*. Retrieved from [https://www.typescriptlang-org.translate.google/?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=es&\\_x\\_tr\\_hl=es-419&\\_x\\_tr\\_pto=sc](https://www.typescriptlang-org.translate.google/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es-419&_x_tr_pto=sc)

Viveros Andrade, S. M., & Sanchez Arce, C. L. J. R. U. y. S. (2018). The academic management of the socio-critical pedagogical model in the educational institution: role of the teacher. *10*(5), 424-433.

What is a Desktop App? (2020). *V2 Cloud*. Retrieved from <https://v2cloud.com/glossary/what-is-a-desktop-app>

## ANEXOS

## Anexo 1: Descripción de Requisitos por procesos.

- Descripción textual del requisito “Listar trámites de registro”

<b>Precondiciones</b>	Debe haber un usuario autenticado con permiso a la funcionalidad.
<b>Flujo de eventos</b>	
<b>Flujo básico Listar trámites de registro</b>	
1.	Se selecciona la funcionalidad Trámites /Trámites Docentes/ Registro.
2.	Se selecciona una de las siguientes opciones: <ul style="list-style-type: none"> <li>• <b>Icono Añadir.</b> Ver_ FORTES_ Levantamiento_AKADEMOS_MINED_DRP_Registrar trámite de registro.odt</li> <li>• <b>Icono Modificar.</b> Ver FORTES_ Levantamiento_AKADEMOS_MINED_DRP_Modificar trámite de registro.odt</li> <li>• <b>Icono Detalles.</b> Ver FORTES_ Levantamiento_AKADEMOS_MINED_DRP_Ver datos de un trámite de registro.odt</li> <li>• <b>Botón Buscar:</b> Ver Flujo alternativo 2.a.</li> <li>• <b>Botón Filtrar:</b> Ver Flujo alternativo 2.b.</li> <li>• <b>Icono Documento del trámite:</b> Se muestra el <b>documento</b> en formato .pdf asociado al <b>Tramite_Docente</b> seleccionado.</li> </ul>
3.	Si selecciona una opción fuera de estas, se concluye el requisito.
<b>Pos-condiciones</b>	
1.	N/A
<b>Flujos alternativos</b>	
<b>Flujo alternativo 2.a Botón Buscar.</b>	
1	Se introduce parámetro para la búsqueda (Nombres y apellidos o Carné de identidad)
2	Se selecciona el botón <b>Buscar</b> .
3	Se busca la coincidencia de los parámetros introducidos con los elementos que se muestran en el Listado de personas. <ul style="list-style-type: none"> <li>- En caso de existir coincidencias, se muestran los elementos del Listado de personas que coincidan.</li> <li>- En caso de que no exista ninguna coincidencia en el Listado de personas, se muestra el mensaje: “ No hay datos”.</li> </ul>
4	Se pasa al Paso 2 del Flujo básico.
<b>Pos-condiciones</b>	
1	N/A
<b>Flujo alternativo 2.b Botón Filtrar.</b>	
1	Al seleccionar al menos un filtro se muestra el panel de filtro con los filtros seleccionados y las opciones siguientes: <ul style="list-style-type: none"> <li>• <b>Mostrar/Ocultar:</b> Ver Flujo alternativo 2.b.a.</li> <li>• <b>Icono Cerrar filtros</b> (imagen de cruz): Ver Flujo alternativo 2.b.b.</li> <li>• <b>Icono Cerrar filtro interno</b> (imagen de signo menos): Ver Flujo alternativo 2.b.c.</li> </ul>
2	Se seleccionan los valores en los filtros.
3	Se muestran los elementos del Listado de personas que coincidan con los valores seleccionados en los filtros. <ul style="list-style-type: none"> <li>- En caso de que no exista ninguna coincidencia en el Listado de personas con los</li> </ul>

	filtros introducidos, se muestra el mensaje: "No hay datos".	
4	Se pasa al Paso 2 del Flujo básico.	
<b>Pos-condiciones</b>		
1	N/A	
<b>Flujo alternativo 2.b.a Mostrar/Ocultar</b>		
1	Se estira o se recoge el panel de filtros, manteniendo seleccionados los filtros y los valores.	
2	Se pasa al Paso 2 del Flujo básico.	
<b>Pos-condiciones</b>		
1	N/A	
<b>Flujo alternativo 2.b.b Cerrar filtros (ícono de cruz)</b>		
1	Se cierran todos los filtros seleccionados.	
2	Se actualiza el Listado de personas quedando como al inicio del requisito.	
3	Se pasa al Paso 2 del Flujo básico.	
<b>Flujo alternativo 2.b.c Cerrar filtro (ícono signo menos)</b>		
1	Se cierra el filtro seleccionado.	
2	Se actualiza el Listado de personas quedando sin la aplicación del o los filtros cerrados.	
3	Se pasa al Paso 2 del Flujo básico.	
<b>Pos-condiciones</b>		
1	N/A	
<b>Validaciones</b>		
1	Tabla Listado de personas	<p>Se muestra con:</p> <ul style="list-style-type: none"> <li>- Las personas que se encuentran en <b>Registro_Estudiente</b> donde el valor <b>tipo_ultimo_trámite</b> es igual a <b>tramite</b> de <b>Tipo_Tramite</b> con valor (Registro), y <b>estado_registro</b> es igual a "1" (abierto).</li> <li>- Y además, que el <b>Registro_Estudiente</b> en el campo <b>centro</b> tengan un valor igual a algunas de las <b>Estructura</b> asociadas al usuario autenticado.</li> </ul> <p>Se muestran inicialmente solo 10 elementos, y luego se muestran dependiendo de la cantidad de elementos que se seleccionen en el componente de paginado.</p> <ul style="list-style-type: none"> <li>- Atributos que se muestran como columnas de la tabla: <ul style="list-style-type: none"> <li>• Foto</li> <li>• Nombres y apellidos</li> <li>• Carné de identidad</li> </ul> </li> </ul> <p>Se muestra organizados alfabéticamente por la columna Nombres y apellidos.</p>
2	Elemento de organización, encabezado de la tabla	Permite que los registros de la tabla se organicen teniendo en cuenta la organización alfabética de los elementos de la columna donde se encuentre.
3	Componente de selección múltiple Filtrar	Muestra los valores del campo de selección de filtros: <ul style="list-style-type: none"> <li>- Provincia</li> <li>- Municipio</li> </ul>
4	Componente de selección simple del filtro Provincia	Muestra todas las Provincias con estado habilitado. Se muestran organizadas alfabéticamente. Al seleccionarse actualiza los elementos del Listado de personas con las <b>Persona</b> donde la <b>direccion</b> contenga la provincia seleccionada.
5	Componente de selección simple del filtro Municipio	Muestra todos los Municipios con estado habilitado que pertenecen a la provincia seleccionada. Se muestran organizados alfabéticamente. Al seleccionarse actualiza los elementos del Listado de personas con las <b>Persona</b> donde la <b>direccion</b> contenga el municipio seleccionado.

		Este filtro se habilita cuando se seleccione un elemento en el componente de filtro Provincia.
<b>Conceptos</b>	<b>Trámite_Docente</b>	<p>Visibles en la interfaz:</p> <ul style="list-style-type: none"> <li>- primer_nombre</li> <li>- segundo_nombre</li> <li>- primer_apellido</li> <li>- segundo_apellido</li> <li>- carne</li> <li>- foto</li> </ul> <p>Utilizados internamente:</p> <ul style="list-style-type: none"> <li>- N/A</li> </ul>
	<b>Estudiante</b>	<p><b>Visibles en la interfaz:</b></p> <ul style="list-style-type: none"> <li>- N/A</li> </ul> <p><b>Utilizados internamente:</b></p> <ul style="list-style-type: none"> <li>- estudiante</li> </ul>
	<b>Registro_Estudiante</b>	<p><b>Visibles en la interfaz:</b></p> <ul style="list-style-type: none"> <li>- N/A</li> </ul> <p><b>Utilizados internamente:</b></p> <ul style="list-style-type: none"> <li>- estudiante</li> <li>- centro</li> <li>- tipo_ultimo_tramite</li> <li>- estado_registro</li> </ul>
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

## Prototipo elemental de interfaz gráfica de usuario

The screenshot displays the 'Estudiantes en registro' page. At the top, there is a search bar with the placeholder text 'Nombre y apellidos, Carné de identidad' and a 'Filtrar' button. Below the search bar, there are two dropdown menus for 'Provincia' and 'Municipio', both currently set to 'Seleccione'. The main content area features a table with the following columns: 'Foto', 'Nombre y apellidos', 'Carné de identidad', and 'Opciones'. The table contains four rows of student records, each with a placeholder for a photo, a full name, a Carné number, and three icons for actions (view, edit, delete). At the bottom of the table, it says 'Mostrando 1 a 10 de 4 registros' and includes pagination controls showing '1' of '2' pages with a '10 / Página' dropdown.

- Descripción textual del requisito “Modificar trámite de registro”

**Precondiciones** Debe haber un usuario autenticado con permiso a la funcionalidad.

	Debe existir al menos una persona con categoría Estudiante y que su último trámite sea: Registro, y su registro debe estar en estado abierto.
<b>Flujo de eventos</b>	
<b>Flujo básico Modificar trámite de registro</b>	
4.	Se muestran los datos del <b>Registro Estudiante</b> seleccionado.
5.	Se introducen o seleccionan los datos para modificar los datos del trámite en la vista Información personal. - Si en los campos se introducen valores inválidos se muestran los mensajes correspondientes debajo del campo y en rojo. Al comenzar a escribir en el campo se elimina el mensaje.
6.	Se selecciona una de las siguientes opciones: <ul style="list-style-type: none"> <li>• <b>Icono Modificar (Persona):</b> Ver Flujo alternativo 3.a.</li> <li>• <b>Botón Cancelar:</b> Ver Flujo alternativo 3.b.</li> <li>• <b>Botón Siguiente:</b> Ver Flujo básico, Paso 4.</li> </ul>
7.	Se valida que los campos obligatorios estén llenos. En caso de no estar llenos, se muestra el mensaje debajo del campo y en rojo: "Campo requerido". Al comenzar a escribir en el campo se elimina el mensaje. Se pasa al Paso 2 del Flujo básico. En caso de estar llenos, se pasa al Flujo básico, Paso 5.
8.	Se selecciona una de las siguientes opciones de la vista Tutores: <ul style="list-style-type: none"> <li>• <b>Buscar:</b> Ver Flujo alternativo 5.a.</li> <li>• <b>Icono Modificar (Persona):</b> Ver Flujo alternativo 3.a.</li> <li>• <b>Icono Añadir:</b> Ver Flujo alternativo 5.b.</li> <li>• <b>Botón Añadir.</b> Ver Flujo alternativo 5.c.</li> <li>• <b>Icono Detalles.</b> Ver flujo alternativo 5.d.</li> <li>• <b>Icono Modificar.</b> Ver flujo alternativo 5.e.</li> <li>• <b>Icono Eliminar.</b> Ver flujo alternativo 5.f</li> <li>• <b>Botón Cancelar:</b> Ver Flujo alternativo 5.g.</li> <li>• <b>Botón Anterior:</b> Ver Flujo alternativo 5.h.</li> <li>• <b>Botón Filtrar:</b> Ver Flujo alternativo 5.i.</li> <li>• <b>Botón Aceptar:</b> Ver Flujo básico, Paso 6.</li> </ul>
9.	- Se actualiza el último trámite del <b>Registro Estudiante</b> , poniendo en <b>Tramite_Docente</b> el <b>estado_tramite</b> con valor igual a "Cancelado".  - Se guarda en <b>Tramite_Docente</b> y <b>Registro</b> un nuevo registro de trámite para el <b>Registro Estudiante</b> , poniendo en <b>Tramite_Docente</b> el <b>tramite</b> igual a <b>tramite</b> del <b>Tramite_Docente</b> que se acaba de cancelar, la <b>fecha_inicio_tramite</b> y la <b>fecha_fin_tramite</b> igual a la fecha actual, y el <b>estado_tramite</b> igual a "Terminado".  - Se actualiza el <b>Registro Estudiante</b> con los valores que vienen de la vista, <b>ultimo_tramite</b> con la referencia del <b>Tramite_Docente</b> que se acaba de registrar y en <b>tipo_ultimo_tramite</b> su tipo.
10.	Se muestra el mensaje: " La acción ha sido realizada satisfactoriamente. " Se pasa FORTES_ Levantamiento_AKADEMOS_MINED_DRP_Listar trámites de registro.odt
<b>Pos-condiciones</b>	
2.	Queda en estado "Cancelado" el último trámite ejecutado para el registro estudiante.
3.	Queda guardado para el registro de estudiante, un nuevo trámite docente del tipo Registro, y el estado del trámite "Terminado".
4.	Queda actualizado el registro de estudiante con los valores asociados al trámite.
<b>Flujos alternativos</b>	
<b>Flujo alternativo 3.a Icono Modificar (Persona).</b>	
1	Se muestra la información de la vista y se introducen o seleccionan los datos para modificar una persona.

	- Si en los campos se introducen valores inválidos se muestran los mensajes correspondientes debajo del campo. Al comenzar a escribir en el campo se elimina el mensaje.
2	Se selecciona una de las siguientes opciones: <ul style="list-style-type: none"> <li>• <b>Botón Cancelar:</b> Ver Flujo alternativo 3.a.a. Botón Cancelar</li> <li>• <b>Botón Aceptar:</b> Ver Flujo alternativo 3.a Icono Modificar (Persona), Paso 3.</li> </ul>
3	Se valida que los campos obligatorios estén llenos. <ul style="list-style-type: none"> <li>• En caso de no estar llenos, se muestra el mensaje debajo del campo y en rojo: "Campo requerido". Al comenzar a escribir en el campo se elimina el mensaje. Se pasa al Paso 1 del Flujo alternativo 3.a Icono Modificar (Persona).</li> <li>• En caso de estar llenos, se al Paso 4 del Flujo alternativo 3.a Icono Modificar (Persona).</li> </ul> <p>Se valida si los datos del campo <b>carné de identidad</b> que se trata de registrar coincide con los datos del <b>carné de identidad</b> de una persona registrada en el sistema.</p> <ul style="list-style-type: none"> <li>• En caso de que coincida, el sistema debe mostrar el siguiente mensaje de información: "El elemento que trata de registrar ya existe", mostrando en color rojo el campo carné de identidad.</li> <li>• En caso de que no coincida, se pasa al Paso 4 del Flujo alternativo 3.a Icono Modificar (Persona).</li> </ul>
4	Se actualiza en el sistema la persona en <b>Persona</b> .
5	Se muestra el mensaje: "La acción ha sido realizada satisfactoriamente." Se pasa a la vista de precedencia.
6	Se actualiza el listado de personas o los datos en el bloque de información Persona, con la persona modificada anteriormente.
<b>Pos-condiciones</b>	
1	Se actualiza en el sistema la persona.
<b>Flujo alternativo 3.a.a Botón Cancelar.</b>	
2	Se muestra el mensaje: ¿Está seguro que desea cancelar la operación?
3	Se selecciona lo opción: <ul style="list-style-type: none"> <li>• <b>Botón Aceptar:</b> Se cierra el mensaje y se pasa a la vista de precedencia.</li> <li>• <b>Botón Cancelar:</b> Se cierra el mensaje y se mantiene en la misma vista.</li> </ul>
<b>Pos-condiciones</b>	
1	N/A
<b>Flujo alternativo 3.b Botón Cancelar.</b>	
1	Se muestra el mensaje: ¿Está seguro que desea cancelar la operación?
2	Se selecciona lo opción: <ul style="list-style-type: none"> <li>• <b>Botón Aceptar:</b> Se cierra el mensaje y se pasa a FORTES_Levantamiento_AKADEMOS_MINED_DRP_Listar trámites de registro.odt</li> <li>• <b>Botón Cancelar:</b> Se cierra el mensaje y se mantiene en la misma vista.</li> </ul>
3	Se pasa al Paso 1 del Flujo básico.
<b>Pos-condiciones</b>	
1	N/A
<b>Flujo alternativo 5.a Buscar.</b>	
1	Se introduce parámetro para la búsqueda (Nombres y apellidos o Carné de identidad)
2	Se selecciona el botón <b>Buscar</b> .
3	Se busca la coincidencia de los parámetros introducidos con los elementos que se muestran en el Listado de personas. <ul style="list-style-type: none"> <li>- En caso de existir coincidencias, se muestran los elementos del Listado de personas.</li> <li>- En caso de que no exista ninguna coincidencia en el Listado de personas, se muestra el mensaje: "No hay datos".</li> </ul>
4	Se pasa al Paso 5 del Flujo básico.



<b>Pos-condiciones</b>	
1	N/A
<b>Flujo alternativo 5.b Icono Añadir.</b>	
1	Se muestra la información de la vista y se introducen o seleccionan los datos para añadir una persona. - Si en los campos se introducen valores inválidos se muestran los mensajes correspondientes debajo del campo. Al comenzar a escribir en el campo se elimina el mensaje.
2	Se selecciona una de las siguientes opciones: <ul style="list-style-type: none"> <li>• <b>Botón Cancelar:</b> Ver Flujo alternativo 5.b.a. Botón Cancelar</li> <li>• <b>Botón Aceptar:</b> Ver Flujo alternativo 5.b Icono Añadir, Paso 3.</li> </ul>
3	Se valida que los campos obligatorios estén llenos. <ul style="list-style-type: none"> <li>• En caso de no estar llenos, se muestra el mensaje debajo del campo y en rojo: " Campo requerido". Al comenzar a escribir en el campo se elimina el mensaje. Se pasa al Paso 1 del Flujo alternativo 5.b Icono Añadir.</li> <li>• En caso de estar llenos, se al Paso 4 del Flujo alternativo 5.b Icono Añadir.</li> </ul> <p>Se valida si los datos del campo <b>carné de identidad</b> que se trata de registrar coincide con los datos del <b>carné de identidad</b> de una persona registrada en el sistema.</p> <ul style="list-style-type: none"> <li>• En caso de que coincida, el sistema debe mostrar el siguiente mensaje de información: " El elemento que trata de registrar ya existe ", mostrando en color rojo el campo carné de identidad.</li> <li>• En caso de que no coincida, se pasa al Paso 4 del Flujo alternativo 5.b Icono Añadir.</li> </ul>
4	Se registra en el sistema una nueva persona en <b>Persona</b> .
5	Se muestra el mensaje: " La acción ha sido realizada satisfactoriamente. " Se pasa a la vista de precedencia (Paso 3 Tutores).
6	Se actualiza el listado de personas que se pueden mostrar en el bloque (Buscar personas) con la persona creada anteriormente.
<b>Pos-condiciones</b>	
1	Se registra en el sistema una nueva persona.
<b>Flujo alternativo 5.b.a Botón Cancelar.</b>	
1	Se muestra el mensaje: ¿Está seguro que desea cancelar la operación?
2	Se selecciona la opción: <ul style="list-style-type: none"> <li>• <b>Botón Aceptar:</b> Se cierra el mensaje y se pasa a la vista de precedencia.</li> <li>• <b>Botón Cancelar:</b> Se cierra el mensaje y se mantiene en la misma vista.</li> </ul>
<b>Pos-condiciones</b>	
1	N/A
<b>Flujo alternativo 5.c Botón Añadir</b>	
1	Si selecciona un elemento del listado de persona, en el bloque que le sigue Datos de tutor se deben poder introducir o seleccionar datos para convertir la persona seleccionada en tutor. Se pasa al Flujo alternativo 5.c Botón Añadir, Paso 2.  Si no selecciona un elemento del listado de personas. El usuario puede ir al paso 5 del flujo básico.
2	El usuario introduce y/o selecciona los datos que le faltan para añadir un tutor.
3	Se valida que los campos estén correctos: <ul style="list-style-type: none"> <li>- Si en los campos se introducen valores inválidos se muestran los mensajes correspondientes debajo del campo. Al comenzar a escribir en el campo se elimina el mensaje.</li> </ul>
4	Se valida que los campos obligatorios estén llenos. <ul style="list-style-type: none"> <li>• En caso de no estar llenos, se muestra el mensaje debajo del campo y en rojo: " Campo requerido". Al comenzar a escribir en el campo se elimina el mensaje. Se pasa al Flujo alternativo 5.c Botón Añadir, Paso 2.</li> <li>• En caso de estar llenos, se habilita el botón Añadir y se pasa al Flujo alternativo 5.c Botón Añadir, Paso 5.</li> </ul>

5	Se selecciona el botón Añadir: <ul style="list-style-type: none"> <li>- El sistema debe validar de que no existan tutores repetidos. <ul style="list-style-type: none"> <li>• En caso de que se intente registrar un tutor que ya existe en ese registro de trámite de matrícula, el sistema debe mostrar el siguiente mensaje: " El elemento que trata de registrar ya existe ", mostrando en color rojo el campo carné de identidad.</li> <li>• En caso contrario el sistema incorpora el tutor al listado de tutores.</li> </ul> </li> </ul>
6	Se pasa al Paso 5 del Flujo básico.
<b>Pos-condiciones</b>	
1	Se registra los datos introducidos para el tutor en <b>Persona</b> .
<b>Flujo alternativo 5.d Icono Detalles</b>	
1	Se selecciona la opción Icono Detalles de las opciones internas del listado de tutores.
2	El sistema debe permitir ver los detalles de un tutor. Permite además seleccionar la siguiente opción: <ul style="list-style-type: none"> <li>• <b>Botón Cerrar:</b> Se cierra la IU y se pasa a la IU de precedencia correspondiente a la sección Tutores.</li> </ul>
<b>Pos-condiciones</b>	
1	N/A
<b>Flujo alternativo 5.e Icono Modificar (Tutor)</b>	
1	El usuario selecciona la opción Icono Modificar en las opciones internas del listado de tutores.
2	El sistema carga los datos del tutor en el bloque Datos de tutor, lo elimina del registro de tutores asociados y lo muestra seleccionado, en la lista de personas del bloque Buscar personas.
3	El usuario introduce y/o selecciona los datos para modificar un tutor.
4	Se valida que los campos estén correctos: <ul style="list-style-type: none"> <li>- Si en los campos se introducen valores inválidos se muestran los mensajes correspondientes debajo del campo. Al comenzar a escribir en el campo se elimina el mensaje.</li> </ul>
5	Se valida que los campos obligatorios estén llenos. <ul style="list-style-type: none"> <li>• En caso de no estar llenos, se muestra el mensaje debajo del campo y en rojo: " Campo requerido ". Al comenzar a escribir en el campo se elimina el mensaje. Se pasa al Flujo alternativo 5.e Icono Modificar (Tutor), Paso 3.</li> <li>• En caso de estar llenos, se habilita el botón Añadir y se pasa al Flujo alternativo 5.e Icono Modificar (Tutor), Paso 6.</li> </ul>
6	Se selecciona el botón Añadir: <ul style="list-style-type: none"> <li>- El sistema debe validar de que no existan tutores repetidos. <ul style="list-style-type: none"> <li>• En caso de que se intente registrar un tutor que ya existe en ese registro de trámite de matrícula, el sistema debe mostrar el siguiente mensaje: " El elemento que trata de registrar ya existe ".</li> <li>• En caso contrario el sistema incorpora el tutor al listado de tutores.</li> </ul> </li> </ul>
7	Se pasa al Paso 5 del Flujo básico.
<b>Pos-condiciones</b>	
1	El sistema registra los datos del tutor en <b>Persona</b> .
<b>Flujo alternativo 5.f Icono Eliminar.</b>	
1	El usuario selecciona la opción <b>Icono Eliminar</b> del listado de registros de tutores.
2	El sistema muestra el mensaje de confirmación: ¿Está seguro que desea eliminar el elemento? Además, permite seleccionar las siguientes opciones: <ul style="list-style-type: none"> <li>- <b>Aceptar.</b> El sistema actualiza el listado de registros de tutores.</li> <li>- <b>Cancelar.</b> Se muestra la misma IU.</li> </ul>
3	Se pasa al Paso 5 del Flujo básico.
<b>Pos-condiciones</b>	
1	Se elimina el registro del elemento seleccionado.

**Flujo alternativo 5.g Botón Cancelar.**

1	Se muestra el mensaje: ¿Está seguro que desea cancelar la operación?
2	Se selecciona la opción: <ul style="list-style-type: none"> <li>• <b>Botón Aceptar:</b> Se cierra el mensaje y se pasa a FORTES_Levantamiento_AKADEMOS_MINED_DRP_Listar trámites de registro.odt</li> <li>• <b>Botón Cancelar:</b> Se cierra el mensaje y se mantiene en la misma vista.</li> </ul>
3	Se pasa al Paso 5 del Flujo básico.

**Pos-condiciones**

1	N/A
---	-----

**Flujo alternativo 5.h Botón Anterior.**

1	Se muestra la vista 2 Información personal, manteniendo los datos introducidos anteriormente.
2	Se pasa al Paso 5 del Flujo básico.

**Pos-condiciones**

1	N/A
---	-----

**Flujo alternativo 5.i Botón Filtrar.**

1	Al seleccionar al menos un filtro se muestra el panel de filtro con los filtros seleccionados y las opciones siguientes: <ul style="list-style-type: none"> <li>• <b>Mostrar/Ocultar:</b> Ver Flujo alternativo 5.i.a.</li> <li>• <b>Icono Cerrar filtros</b> (imagen de cruz): Ver Flujo alternativo 5.i.b.</li> <li>• <b>Icono Cerrar filtro interno</b> (imagen de signo menos): Ver Flujo alternativo 5.i.c.</li> </ul>
2	Se seleccionan los valores en los filtros.
3	Se muestran los elementos del Listado de personas que coincidan con los valores seleccionados en los filtros. <ul style="list-style-type: none"> <li>- En caso de que no exista ninguna coincidencia en el Listado de personas con los filtros introducidos, se muestra el mensaje: "No hay datos".</li> </ul>
4	Se pasa al Paso 5 del Flujo básico.

**Pos-condiciones**

1	N/A
---	-----

**Flujo alternativo 5.i.a Mostrar/Ocultar**

1	Se estira o se recoge el panel de filtros, manteniendo seleccionados los filtros y los valores.
2	Se pasa al Paso 5 del Flujo básico.

**Pos-condiciones**

1	N/A
---	-----

**Flujo alternativo 5.i.b Cerrar filtros (ícono de cruz)**

1	Se cierran todos los filtros seleccionados.
2	Se actualiza el Listado de personas quedando como al inicio del requisito.
3	Se pasa al Paso 5 del Flujo básico.

**Pos-condiciones**

1	N/A
---	-----

**Flujo alternativo 5.i.c Cerrar filtro (ícono signo menos)**

1	Se cierra el filtro seleccionado.
2	Se actualiza el Listado de personas quedando sin la aplicación del o los filtros cerrados.
3	Se pasa al Paso 5 del Flujo básico.

**Pos-condiciones**

1	N/A
---	-----

**Validaciones**

6	Tabla Listado de personas (Tutores)	Bloque Buscar personas: <ul style="list-style-type: none"> <li>- Se muestra inicialmente vacía.</li> <li>- Al solicitar búsqueda de información sobre ella, muestra: <ul style="list-style-type: none"> <li>• Las personas registradas en el sistema excepto el estudiante al que se le está</li> </ul> </li> </ul>
---	--	---

		<p>intentando modificar el trámite de registro.</p> <ul style="list-style-type: none"> <li>- Se muestran inicialmente solo 10 elementos, y luego se muestran dependiendo de la cantidad de elementos que se seleccionen en el componente de paginado.</li> <li>- Atributos que se muestran como columnas de la tabla: <ul style="list-style-type: none"> <li>• Foto</li> <li>• Nombres y apellidos</li> <li>• Carné de identidad</li> </ul> </li> <li>- Se muestra organizados alfabéticamente por la columna Nombres y apellidos.</li> <li>- El usuario debe poder seleccionar un solo elemento del listado de personas.</li> </ul> <p>Bloque Tutores asociados:</p> <ul style="list-style-type: none"> <li>- La tabla se muestra con los tutores registrados en el registro del trámite de registro.</li> <li>- Atributos que se muestran como columnas de la tabla: <ul style="list-style-type: none"> <li>• Foto</li> <li>• Nombre y apellidos</li> <li>• Carné de identidad</li> <li>• Parentesco</li> <li>• Formalizado</li> </ul> </li> </ul>
7	Elemento de organización, encabezado de la tabla	Permite que los registros de la tabla se organicen teniendo en cuenta la organización alfabética de los elementos de la columna donde se encuentre.
8	Componente de selección múltiple Filtrar	Muestra los valores del campo de selección de filtros: <ul style="list-style-type: none"> <li>- Provincia</li> <li>- Municipio</li> </ul>
9	Componente de selección simple del filtro Provincia	Muestra todas las Provincias con estado habilitado. Se muestran organizadas alfabéticamente. Al seleccionarse actualiza los elementos del Listado de personas con las <b>Persona</b> donde la <b>direccion</b> contenga la provincia seleccionada.
10	Componente de selección simple del filtro Municipio	Muestra todos los Municipios con estado habilitado que pertenecen a la provincia seleccionada. Se muestran organizados alfabéticamente. Al seleccionarse actualiza los elementos del Listado de personas con las <b>Persona</b> donde la <b>direccion</b> contenga el municipio seleccionado. Este filtro se habilita cuando se seleccione un elemento en el componente de filtro Provincia.
11	Panel de información de Información del trámite	Se muestran los valores correspondientes al <b>Registro_Estudiente</b> y a <b>Persona</b> .
12	Componente de selección simple Situación escolar	<ul style="list-style-type: none"> <li>- Si el valor de <b>cambio_situacion</b> es "1" del <b>Tipo_Trámite</b> que se realiza (Registro); se muestra habilitado el campo Situación escolar con los valores de <b>situacion</b> asociados a <b>Tipo_Trámite</b>.</li> <li>- Si el valor de <b>cambio_situacion</b> es "0" del <b>Tipo_Trámite</b> que se realiza (Registro); se muestra el campo Situación escolar inhabilitado.</li> <li>- Se muestran ordenadas alfabéticamente.</li> </ul>
13	Componente de selección simple Provincia	El componente de selección simple Provincia debe mostrar los valores del atributo <b>provincia</b> existentes en el sistema y con estado habilitado. Se muestran ordenadas alfabéticamente.
14	Componente de selección simple Municipio	El componente de selección simple Municipio debe mostrar los valores del atributo <b>municipio</b> existentes en el sistema, con estado habilitado y

		en dependencia de la provincia seleccionada previamente en el componente de selección simple Provincia. Este campo se habilita cuando se seleccione un elemento en el campo Provincia. Se muestran ordenadas alfabéticamente.
15	Componente de selección simple Localidad	El componente de selección simple Localidad debe mostrar los valores del atributo <b>localidad</b> existentes en el sistema, con estado habilitado y en dependencia del municipio seleccionado previamente en el componente de selección simple Municipio. Este campo se habilita cuando se seleccione un elemento en el campo Municipio. Se muestran ordenadas alfabéticamente.
16	Componente de selección simple Consejo popular	El componente de selección simple Consejo popular debe mostrar los valores del atributo <b>consejo_popular</b> existentes en el sistema, con estado habilitado y en dependencia de la localidad seleccionada previamente en el componente de selección simple Localidad. Este campo se habilita cuando se seleccione un elemento en el campo Localidad. Se muestran ordenadas alfabéticamente.
17	Componente de selección simple Vinculación laboral	<ul style="list-style-type: none"> <li>- Si en el componente se seleccionan los elementos; Al sector estatal o Al sector no estatal, deben habilitarse los campos: Centro de trabajo, Sector laboral y Procedencia social todos con carácter obligatorio.</li> <li>- Si en el componente se selecciona; Sin vínculo laboral deben, deshabilitarse los campos: Centro de trabajo, Sector laboral y Procedencia social.</li> <li>- Inicialmente los campos Centro de trabajo, Sector laboral y Procedencia social se muestran deshabilitados.</li> </ul>
18	Componente checkbox Formalizado	Debe mostrarse inicialmente habilitado y marcado.
<b>Conceptos</b>	<b>Persona</b>	<p>Visibles en la interfaz:</p> <ul style="list-style-type: none"> <li>- primer_nombre</li> <li>- segundo_nombre</li> <li>- primer_apellido</li> <li>- segundo_apellido</li> <li>- carne</li> <li>- no_serie_ci</li> <li>- fecha_nacimiento</li> <li>- dirección</li> <li>- municipio</li> <li>- provincia</li> <li>- consejo_popular</li> <li>- localidad</li> <li>- teléfono 1</li> <li>- teléfono 2</li> <li>- correo</li> <li>- sexo</li> <li>- raza</li> </ul> <p>Utilizados internamente:</p> <ul style="list-style-type: none"> <li>- N/A</li> </ul>
	<b>Tramite_docente</b>	<p>Visibles en la interfaz:</p> <ul style="list-style-type: none"> <li>- situacion_escolar</li> </ul> <p>Utilizados internamente:</p>

		<ul style="list-style-type: none"> <li>- tramitador</li> <li>- fecha_inicio_tramite</li> <li>- fecha_fin_tramite</li> <li>- estado_tramite</li> <li>- tramite</li> <li>- curso_escolar</li> </ul>
	<b>Tutor_Legal</b>	<p>Visibles en la interfaz:</p> <ul style="list-style-type: none"> <li>- parentesco</li> <li>- vinculación_laboral</li> <li>- procedencia_social</li> <li>- centro_trabajo</li> <li>- sector_laboral</li> <li>- formalizado</li> <li>- descripcion</li> </ul> <p>Utilizados internamente:</p> <ul style="list-style-type: none"> <li>- N/A</li> </ul>
	<b>Estudiante</b>	<p>Visibles en la interfaz:</p> <ul style="list-style-type: none"> <li>- huérfano</li> <li>- nacionalidad</li> </ul> <p>Utilizados internamente:</p> <ul style="list-style-type: none"> <li>- N/A</li> </ul>
	<b>Registro_estudiante</b>	<p>Visibles en la interfaz:</p> <ul style="list-style-type: none"> <li>- N/A</li> </ul> <p>Utilizados internamente:</p> <ul style="list-style-type: none"> <li>- situación_escolar</li> <li>- estado_docente</li> <li>- ultimo_tramite</li> <li>- tipo_ultimo_tramite</li> <li>- estado_registro</li> </ul>
	<b>Tramite_Registro</b>	<p>Visibles en la interfaz:</p> <ul style="list-style-type: none"> <li>- fecha</li> </ul> <p>Utilizados internamente:</p> <ul style="list-style-type: none"> <li>- tramite_docente</li> </ul>
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

### Prototipo elemental de interfaz gráfica de usuario

**Modificar trámite de registro**

Información personal Tutores

**Buscar personas**

Nombre y apellidos, Carné de identidad

Mostrar/Ocultar

Provincia:  Municipio:

Foto	Nombre y apellidos	Carné de identidad	Opciones
<input type="radio"/>	Nombre 1 Nombre 2 Apellido 1 Apellido 2	93022201613	
<input type="radio"/>	Nombre 1 Nombre 2 Apellido 1 Apellido 2	94052201613	
<input checked="" type="radio"/>	Nombre 1 Nombre 2 Apellido 1 Apellido 2	98022201613	<input type="button" value="✎"/>
<input type="radio"/>	Nombre 1 Nombre 2 Apellido 1 Apellido 2	99022201613	

Mostrando 1 a 10 de 4 registros 1 2 > 10 Página

**Datos de tutor**

Parentesco:  Vinculación laboral:  Centro de trabajo:

Procedencia social:  Sector laboral:   Formalizado

Descripción:

**Tutores asociados**

Foto	Nombre y apellidos	Carné de identidad	Parentesco	Formalizado	Opciones
<input type="checkbox"/>	Nombre 1 Nombre 2 Apellido 1 Apellido 2	93022201613	Padre	Si	<input type="button" value="📄"/> <input type="button" value="✎"/> <input type="button" value="🗑"/>
<input type="checkbox"/>	Nombre 1 Nombre 2 Apellido 1 Apellido 2	94052201613	Madre	Si	<input type="button" value="📄"/> <input type="button" value="✎"/> <input type="button" value="🗑"/>

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

## Anexo 2: Acta de Aceptación del Producto.

## Anexo 3: Encuesta inicial de la investigación.

Especialista, le invito a responder el siguiente cuestionario con el fin de conseguir su colaboración en la presente investigación, solicito que exprese en sus respuestas criterios verídicos que guíen al autor de la investigación. Marque con una X en una sola opción y en el caso de la 5 responda brevemente. Por el tiempo brindado, muchas gracias.

1. ¿Considera importante la implementación de una aplicación de escritorio sin conexión a internet que permita el uso del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED?

Sí \_\_\_ No \_\_\_ No sé \_\_\_

2. ¿Considera usted correcta la forma en que se brindan los servicios actualmente a los usuarios?

Sí \_\_\_ No \_\_\_ No sé \_\_\_

3. ¿Considera usted factible la creación de una aplicación de escritorio sin conexión a internet que permita el uso del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED?

Sí \_\_\_ No \_\_\_ No sé \_\_\_

4. Luego de haber mostrado los resultados de la solución refleje en qué medida le gusta la solución desarrollada.

\_\_\_Me gusta mucho                      \_\_\_Me disgusta más de lo que me gusta

\_\_\_Me gusta más de lo que me disgusta                      \_\_\_No me gusta nada

Me da lo mismo

\_\_\_No sé decir

5. ¿Qué opina usted acerca de los beneficios que traería para las escuelas del país el desarrollo de una aplicación de escritorio sin conexión a internet que permita el uso del Sistema de Gestión Académica XAUCE AKADEMOS para el MINED?