



Universidad de las Ciencias Informáticas
Facultad 1

Sistema de Gestión de Estudiantes para el Sector de Apoyo Estudiantil
de Angola en Cuba.

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

Autor: Pedro Massengo Nekaka

Tutoras:

MSc. Maidelis Milanés Luque

Ing. Liliet de la Caridad González Polanco

Ciudad de La Habana, junio de 2020

“Año 62 de la Revolución”

DATOS DE CONTACTO

Tutoras:

Maidelis Milanés Luque: Graduada de Ingeniera en Ciencias Informáticas en el 2007, Máster en informática aplicada en el 2017, actualmente realizando investigaciones en el área de la Inteligencia Artificial, específicamente en Algoritmos de Agrupamiento Conceptual, utilizando el enfoque lógico combinatorio y las Redes Neuronales Artificiales. Profesora de la disciplina de Inteligencia Artificial.

Correo: mmilanes@uci.cu

Liliet de la Caridad González Polanco: Graduada de Ingeniera en Ciencias Informáticas en el 2017 en la Universidad de las Ciencias Informáticas (UCI). Maestrante de la Maestría de Calidad de Software acreditada de Excelencia por la Junta de Acreditación Nacional. Profesora de la Disciplina de Inteligencia Computacional y Especialista Comercial de la Dirección de Transferencia de Tecnología de la UCI.

Correo: lcgonzalez@uci.cu

AGRADECIMIENTOS

Primeramente, agradezco a Dios Jehová, a Jesús Cristo mi salvador y señor, al Espíritu Santo el consolador. Agradezco a mi familia mi esposa, mis padres, mis hermanos, primos y tíos. Agradezco al gobierno angoleño por la oportunidad que me fue dada y a la universidad UCI por recibirme y cuidarme en mi estancia en Cuba. También a mis dos tutoras queridas y a todos que contribuyeron directa o indirectamente en mi formación académica y en mi personalidad.

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo Pedro Massengo Nekaka, soy el autor principal del trabajo titulado “Sistema de Gestión de Estudiantes para el Sector de Apoyo Estudiantil de Angola en Cuba”, por lo que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmo la presente a los _____ días del mes de _____ del año 2020.

Autor:

Pedro Massengo Nekaka

Tutores:

MSc. Maidelis Milanés Luque

Ing. Liliét de la C. González Polanco

RESUMEN

Como parte del plan de formación de profesionales angoleños, el gobierno permite el intercambio universitario con otros países. Para la gestión de la información de estos estudiantes, el Sector de Apoyo Estudiantil (SAE) que es encargado del seguimiento y atención a los estudiantes en el país destino, utiliza una base de datos Access que no permite la gestión de los procesos relacionados con el registro de los graduados, el registro de estudiantes dados en baja ni su proceso de reingreso.

La presente investigación propone un Sistema de Gestión de Estudiante (SGE) para contribuir en la gestión de la información de estudiantes angoleños en Cuba, haciendo uso de la metodología de desarrollo de software XP, el marco de trabajo MERN (Mondo db, Express, React JS y Node JS, JavaScript como lenguaje de programación y como editor de texto Visual Studio Code. La propuesta fue validada a partir de las pruebas de aceptación y unitarias, se obtuvo como resultado un sistema informático web que cumple con los requisitos deseados

Palabras clave: gestión de la información, sistema de gestión estudiantil, sistema informático web, SAE.

ÍNDICE

INTRODUCCIÓN	1
Estructura organizativa del documento	4
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	6
1.1 Conceptos asociados a la investigación.....	6
1.2 Descripción de los procesos asociados a la gestión de la información de estudiantes en el Sector de Apoyo Estudiantil en Cuba	7
1.3 Estudio de sistemas similares	12
1.4 Metodología de Desarrollo de Software	16
1.4.1 Selección de tipo de la Metodología.....	16
1.4.2 Comparación de Metodología Ágiles.....	16
1.5 Tecnologías, lenguaje de programación y herramientas	18
1.6 Herramientas a utilizar	22
1.7 Conclusiones del capítulo	23
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA	25
2.1 Descripción de la propuesta de la Solución	25
2.2 Usuarios del Sistema.....	26
2.3 Planificación.....	26
2.3.1 Lista de reserva del producto	26
2.3.2 Historias de Usuario	27
2.3.3 Requisitos no funcionales.....	31
2.3.4 Estimación de Esfuerzo por Historias de Usuario.....	32
2.3.5 Plan de Iteraciones para entregables	34
2.4 Diseño de la Aplicación	35
2.4.1 Tarjetas CRC	35

2.4.2 Arquitectura.....	37
2.4.2.1 Patrón Arquitectónico.....	38
2.4.3 Patrones de Diseño.....	39
2.5 Conclusiones del capítulo.....	41
CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA	42
3.1 Desarrollo	42
3.1.1 Modelo de Datos	42
3.1.2 Tareas de Ingeniería	44
3.2 Prueba.....	48
3.2.1 Pruebas Unitarias.....	48
3.2.2 Pruebas de Aceptación.....	50
3.2.3 Pruebas de Seguridad	54
3.3 Implementación y validación del sistema	55
3.4 Conclusiones del capítulo	56
CONCLUSIONES GENERAL.....	¡Error! Marcador no definido.
RECOMENDACIONES.....	58
REFERENCIA BIBLIOGRAFICA	59
ANEXO.....	65
Anexo 1: Historias de usuarios.....	65
Anexo 2: Tarjetas CRC.	71
Anexo 3: Tarea d Ingeniería.	74
Anexo 4: Prueba de Aceptación.	82
Anexo 5: Entrevista.	95

INTRODUCCIÓN

El uso de las Tecnologías de la Información y las Comunicaciones (TIC) constituye el exponente más alto en las instituciones modernas. Están presentes en diversos sectores tales como: educación, salud, comunicación y en el mundo empresarial de forma general. Debido a la información que es recopilada, es necesario un método adecuado y que se cumplan los pilares de la seguridad (ISO 27001): **integridad** (garantiza que los datos no sean modificados desde su creación sin autorización, se debe asegurar que ningún intruso pueda capturar y modificar los datos en tránsito), **confidencialidad** (avala que la información almacenada en el sistema informático o transmitida por la red solamente esté disponible para aquellas personas autorizadas a accederla), **disponibilidad** (garantiza el correcto funcionamiento de los sistemas de información y su disponibilidad en todo momento para los usuarios autorizados) y **autenticación** (asegura que sólo los individuos autorizados tengan acceso a los recursos) [1].

El gobierno de Angola en su plan de formación de profesionales para el país crea el Instituto Nacional de Gestión de Becas de Estudio (INAGBE), esta institución es la encargada de gestionar las becas internas (asignar becas a los estudiantes en las distintas universidades del país) y externas (las becas son asignadas para las universidades de distintos países). Cualquier estudiante que solicita auxilio financiero en su formación de pregrado o postgrado, debe pasar en un proceso en que se verifican los resultados obtenidos en las enseñanzas precedentes para comprobar que el promedio se encuentra en el rango establecido, también se analiza la condición financiera del solicitante (pues la ayuda es para personas con dificultad financiera). Según consta en la declaración de compromiso es deber del becado: tener comportamiento moral y cívico irreprochable, tener buenos resultados académicos, no abandonar la carrera antes del final previsto y en caso de ser becado externo debe regresar al país para contribuir en el desarrollo de este. Al incumplir el reglamento, se compromete la beca y en casos extremos se incurre en la devolución del presupuesto invertido hasta el momento [2].

Como representante de esta institución en el exterior surge el SAE, organismo competente encargado del seguimiento y atención a los estudiantes que cursan actividades de pregrado y postgrado. Un estudio realizado en el Sector de Apoyo a los Estudiantes en Cuba muestra la relación entre esta entidad y las entidades del país acogedor; el SAE en Cuba contacta con el Ministerio de Enseñanza Superior (MES) para solicitar plazas en las universidades y proceder al pago anual de las carreras cursadas por los estudiantes, también contacta con el Ministerio de Salud Pública (MINSAP) para verificar las cuestiones referentes a los

seguros de salud. Las universidades, al finalizar cada semestre, envían al SAE un informe con la conducta y los resultados académicos obtenidos por cada estudiante. La gestión de la información estudiantil en el SAE se hace mediante los siguientes procesos: registrar los estudiantes, registrar el resultado académico individual del año culminado, registrar las universidades en que los estudiantes viven y estudian, y el estipendio del año. Estos procesos se llevan a cabo haciendo uso de una base de datos Access, una herramienta de la empresa Microsoft, donde se almacena toda la información referente a los estudiantes.

Las limitaciones de almacenamiento que presenta la base de datos Access, conlleva a que sean eliminados los datos de los estudiantes ya graduados y la trayectoria académica de los estudiantes en cursos anteriores, lo que ocasiona pérdida de información importante que puede ser utilizada en futuros análisis, favoreciendo la toma de decisiones. Además, también son eliminados los datos de los estudiantes que causan baja académica, por lo que, si un estudiante tiene la posibilidad de reingresar se deben recoger nuevamente los datos académicos. Al mismo tiempo, la existencia de tablas no normalizadas origina dificultad en la actualización de los datos. Por otro lado, el sistema no permite la gestión de los procesos relacionados con el registro de graduados, de estudiantes dados de baja académica, de reintegro de estudiantes y la gestión de resultados académicos. También se debe tener en cuenta que Access no es una herramienta libre de uso, por lo que causa gastos económicos adicionales para pagar la licencia.

Teniendo en cuenta la importancia que representan la correcta gestión de esta información para el trabajo común del sector, se plantea como **problema de investigación**: ¿Cómo contribuir a la gestión de la información de estudiantes sin perder la información almacenada en el Sector de Apoyo Estudiantil de Angola en Cuba?

El problema identificado enmarca como **objeto de estudio** el proceso de gestión de la información.

Para dar solución al problema descrito se propone el siguiente **objetivo general**: Desarrollar el Sistema de Gestión de Estudiante para contribuir en la gestión de la información de estudiantes en el Sector de Apoyo Estudiantil de Angola en Cuba.

Se identifica como **campo de acción**: gestión de la información de estudiantes en el Sector de Apoyo Estudiantil en Cuba.

Para dar solución al problema descrito se formularon los siguientes **objetivos específicos**:

- Definir el marco teórico de la investigación mediante el estudio y análisis de los principales referentes teóricos para el desarrollo de la solución, asociados a la gestión de la información de estudiantes en el Sector de Apoyo Estudiantil en Cuba.
- Realizar el diseño e implementación del sistema de gestión de estudiantes para el SAE, según la metodología, tecnologías y herramientas seleccionadas.
- Realizar pruebas para validar el correcto funcionamiento de la solución propuesta.

En aras de concretar el objetivo planteado se definen las siguientes **tareas de investigación**:

1. Descripción de los procesos de negocio asociados a la gestión de la información de estudiantes en el Sector de Apoyo Estudiantil en Cuba.
2. Realización de un estudio del estado del arte, para conocer los referentes teóricos del proceso de gestión de la información.
3. Selección de la metodología de desarrollo software y las tecnologías necesarias para la implementación de la propuesta de solución.
4. Caracterización de las herramientas de desarrollo adecuadas para la implementación del software.
5. Identificación de los elementos necesarios para el análisis y diseño del Sistema de Gestión de Estudiante para el Sector de Apoyo Estudiantil en Cuba, según la metodología de desarrollo de software escogida.
6. Identificación de los elementos necesarios para la implementación del Sistema de Gestión de Estudiante para el Sector de Apoyo Estudiantil en Cuba, según la metodología de desarrollo de software escogida.
7. Implementación y validación de la solución propuesta a través de pruebas de software.

Como **Hipótesis** se plantea: el desarrollo de un sistema informático para la gestión estudiantil en el Sector de Apoyo Estudiantil de Angola en Cuba, contribuirá a la gestión de la información de los procesos asociados a los estudiantes y a la persistencia de la información almacenada.

Se utilizaron los siguientes **métodos teóricos**:

- **Histórico – lógico**: se utiliza este método para determinar los antecedentes, tendencias, y regularidades del proceso de elaboración de la gestión de la información de estudiantes en el Sector de Apoyo Estudiantil en Cuba, sustentado en las tecnologías y las herramientas informáticas para la

gestión de la información.

- **Analítico – sintético:** se utiliza este método para el estudio de la base teórica, principalmente de las herramientas y tecnologías a emplear, para definir las características esenciales de las mismas y sus relaciones. Además, se utilizó para estudiar los procesos, los sistemas similares e identificar las posibles funcionalidades de la propuesta de solución.
- **Modelación:** se utiliza este método durante el desarrollo de la investigación para presentar la propuesta de solución, así como en la generación del modelo de datos, arquitectura y otros tipos de diagramas relacionados con el sistema según la metodología de desarrollo de software.

Se emplean los siguientes **métodos empíricos**:

- **Observación:** se utiliza este método para obtener de forma directa, la información de la realidad objetiva del comportamiento del proceso de gestión de la información de estudiantes sustentado en las tecnologías, en el Sector de Apoyo Estudiantil en Cuba.
- **Entrevistas:** se utiliza este método para realizar una entrevista estructurada al responsable del Sector de Apoyo Estudiantil en Cuba, con el objetivo de obtener información referente a los pormenores en el proceso de gestión de la información estudiantil, las principales dificultades que presenta este proceso y las características más importantes que debían ser incluidas en la solución.

Estructura organizativa del documento

La tesis está organizada por una introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas, anexos. A continuación, se describe el contenido propuesto en cada capítulo.

Capítulo I: “Fundamentación Teórica”, se especifican los conceptos relacionados con la gestión de la información de estudiantes, lo que permitirá la comprensión del dominio del problema. Se hace un análisis de sistemas similares a la temática de la investigación. Se realiza una descripción de la metodología de desarrollo de software, las tecnologías, el lenguaje de programación, el marco y las herramientas que serán utilizadas para el desarrollo de la solución propuesta.

Capítulo II: “Características del Sistema”, se describe la propuesta de solución, tomando en cuenta los elementos que se definen en la fase de Exploración según la metodología XP. Se detallan las Historias de

Usuario que recogen todas las funcionalidades a implementar. Además, se muestran otros artefactos propios de la metodología de desarrollo seleccionada.

Capítulo III: “Implementación y Pruebas”, se define el modelo de datos, las pautas de la implementación, así como la validación de la propuesta de la solución a través de pruebas de software.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se presentan los conceptos fundamentales de la investigación y un estudio de los sistemas similares, con el fin de lograr una mejor comprensión y adquirir conocimientos referentes a los sistemas de gestión de información estudiantil. Además, se presenta la metodología y el ambiente de desarrollo que se emplearán en la implementación de la propuesta de solución.

1.1 Conceptos asociados a la investigación

En todo sistema de gestión de información, el elemento principal es la información, por lo que es necesario definirla y caracterizarla. Según la Real Academia de la Lengua Española, se define la **información** como la comunicación o adquisición de conocimientos que permiten ampliar o precisar los que se poseen sobre una materia determinada [3]. Esta definición hace referencia a la utilidad de la información como fuente de conocimientos.

Otro concepto con el que se define a la información es en el ámbito de Sistemas de Información, en el cual Börje Langefors define la información como “cualquier mensaje o conocimiento que pueda usarse para posibilitar o mejorar una acción o decisión” [4], mencionando el valor de la información en la toma de decisiones en cualquier organización.

Antes de hablar de la **Gestión de la Información** (GI), es necesario tener en cuenta el concepto de gestión que es “la actividad dirigida a obtener y asignar los recursos necesarios para el cumplimiento de los objetivos de la organización.” Varios son los autores que coinciden con que la información necesita una buena gestión, o sea, obtenerla de una fuente confiable, analizarla, procesarla y utilizarla en la toma de decisiones de una organización. Según Phil Bartle, “la gestión de la información es el proceso de analizar y utilizar la información que se ha recabado y registrado para permitir a los administradores (de todos los niveles) tomar decisiones documentadas” [5].

Aterrizando la definición antes mencionada desde el punto de vista de la ingeniería, se puede decir que la GI, se pudiera entender por “el proceso de extracción, manipulación, tratamiento, depuración, conservación y acceso a la información obtenida a través de una o varias fuentes, además de gestionar los derechos de los usuarios a usarla” [6].

En el mundo, el desarrollo de las tecnologías va aparejado al incremento de sistemas para la gestión de la información en las distintas esferas. Estos constituyen una estructura probada para la gestión y mejora

continua de las políticas, procedimientos y procesos de toda organización. La necesidad de analizar y comprender dichos sistemas, tanto en el ámbito nacional, como internacional es el factor clave para la realización de un nuevo producto informático. **Los Sistemas de Gestión** deben facilitar, simplificar y realizar automáticamente procesos que tradicionalmente se realizaban de forma manual [6]. Así pues, sustituyen ventajosamente al personal encargado, evitando errores y mejorando la velocidad de procesamiento de la información; establecen un progresivo control en las entidades financieras con ventajas incuestionables en cuanto a fiabilidad y seguridad; realizan los reiterativos procesos contables sin errores en las operaciones y con una rapidez y agilidad considerable.

Un **Sistema Informático** puede ser definido como un sistema de información que basa la parte fundamental de su procesamiento, en el empleo de la computación, como cualquier sistema, es un conjunto de funciones interrelacionadas, hardware, software y de recurso humano. Un sistema informático normal emplea un sistema que usa dispositivos que se usan para programas y datos [7].

Una **Base de Datos o Banco de Datos (BB.DD)**, es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta [8].

1.2 Descripción de los procesos asociados a la gestión de la información de estudiantes en el Sector de Apoyo Estudiantil en Cuba

Para entender la implementación y el desarrollo de la Educación Superior en Angola, es importante recurrir a la historia, desde el año 1962, con la creación de Estudios Generales Universitarios (EGU), que más tarde evolucionó y adquirió el estatus universitario en virtud del Decreto Ley 48/90, de 23 de diciembre de 1968, que adoptó la designación de la Universidad de Luanda [9].

El EGU comenzó sus actividades docentes efectivamente un año tras su fundación, inicialmente se priorizaron las carreras de Medicina e Ingeniería en la provincia de Luanda; Medicina Veterinaria, Agronomía, en Huambo; Silvicultura y Ciencias Pedagógicas, en la Huíla [9]. Con la proclamación de la independencia colonial en el año 1975, el gobierno angolano decide fortalecer el capital humano, como primer paso procede a la creación de universidades donde cualquier ciudadano, sin distinción alguna, pudiera libremente ejercer sus estudios de pregrado. Tal medida es imprescindible para el desarrollo de un

país, pues así se logra la verdadera libertad, como refiere José Martí: “*Ser culto es el único modo de ser libre*”.

Actualmente, Angola cuenta con cerca de 29 instituciones de educación superior pública distribuidas en todas las provincias del país, donde se especializan a cada año un número significativo de profesionales [10]. A continuación, se ilustra una tabla con el listado de todas las universidades del país, la misma muestra su ubicación y el año de fundación.

Tabla 1. Universidades Públicas de Angola [9].

Sector Académico	#	Nombre de la Universidad	Año de fundación	Ubicación
I	1	Universidade Agostinho Neto	1962	Luanda
	2	Instituto Superior de Serviço Social	2009	Luanda
	3	Instituto Superior de Tecnologia de Informação e Comunicação	2012	Luanda
	4	Instituto Superior de Artes	2009	Luanda
	5	Instituto Superior de Educação Física e Desportos	2013	Luanda
	7	Instituto Superior de Ciências da Comunicação	2009	Luanda
	8	Instituto Superior Ciências da Educação de Luanda	2009	Luanda
	II	9	Universidade Katyavala Buila	2009
10		Instituto Superior Politécnico do Cuanza Sul	2009	Cuanza Sul
11		Instituto Superior de Petróleos	2009	Cuanza Sul
III	12	Universidade 11 de Novembro	2009	Cabinda y Zaire

IV	13	Universidade Lueji A´Nkonde	2009	Malange, Lunda Norte, Lunda Sul
	14	Escola Superior Politécnica de Malanje	2009	Malanje
	15	Instituto Superior Politécnico de Malanje	2009	Malanje
	16	Escola Superior Pedagógica da Lunda Sul	2009	Lunda Sul
	17	Instituto Superior Técnico Agro-Alimentar de Malanje	2009	Malanje
V	18	Universidade Eduardo dos Santos	2009	Huambo, Bie, Moxico
	19	Instituto Superior de Ciências de Educação do Huambo	2009	Huambo
	20	Escola Superior Pedagógica do Bié	2009	Bié
VI	21	Universidade Mamdume Ya Ndemufaio	2009	Huíla, Namibe
	22	Instituto Superior de Pescas	2009	Namibe
	23	Instituto Superior de Ciências de Educação de Lubango	2009	Huíla
VII	24	Universidade Kimpa Vita	2009	Uíge, Kwanza Norte
	25	Escola Superior Pedagógica do Cuanza Norte	2009	Kwanza Norte
	26	Instituto Superior Politécnico do Cuanza Norte	2009	Kwanza Norte
	27	Instituto Superior de Ciências de Educação do Uíge	2009	Uíge

	28	Academia de Ciências da Saúde	2009	Uíge
VIII	29	Universidade Cuíto Cuanavale	2009	Cunene, Cuando Cubango

Con la necesidad de facilitar la formación de profesionales surge INAGBE el 08 de mayo de 1980, mientras Angola aún vivía tiempos difíciles de la guerra. INAGBE tiene la misión de crear condiciones para ayudar a los ciudadanos angoleños que deseen proseguir sus estudios, pero con escasa condición económica para alcanzar dichos propósitos o para aquellos que quieren estudiar en otro país.

Existen dos procesos distintos para la obtención de beca, uno para lo que deseen estudiar en el país y otro correspondiente a los que desean estudiar en un país extranjero.

El proceso para los que quieren estudiar en el país es:

- INAGBE hace un llamado en marzo, pues es el mes en que las clases dan su inicio en la enseñanza superior, informando la capacidad disponible de estudiantes a los cuáles la institución podrá costearle sus estudios. Para anotarse es necesario que el estudiante se encuentre matriculado en una universidad pública o privada y no sea beneficiario de una beca de otra institución.
- Los estudiantes deben dirigirse a las instalaciones del INAGBE, llevando la declaración que confirma que ya están matriculados en una universidad, sus documentos de identidad y los certificados académico del año culminado con las notas.
- Con ayuda de un sistema informático, INAGBE registra el estudiante y hace la selección de los estudiantes. Si el estudiante es aceptado, será notificado mediante la universidad donde se encuentra matriculado, quedando a partir de ese momento exento del pago de sus estudios y tendrán que hacer trámites con el INAGBE y un banco donde recibirán un estipendio. Los estudiantes no pueden desaprobado un semestre para mantener la beca activa.

El proceso para los que quieren estudiar en el extranjero es:

- INAGBE hace un llamado en cualquier momento del año, dependiendo de las negociaciones hechas con los países disponibles a recibir estudiantes y según el presupuesto que el país pone a disposición. Se les informa a los aspirantes la capacidad disponible y el país destino. Los candidatos no pueden ser beneficiarios de becas de otras instituciones.

- Los candidatos deben dirigirse a las instalaciones del INAGBE con toda la documentación requerida. Además, deben cumplir los siguientes requisitos: edad entre los 17 a los 22, media académica de los años anteriores superior o igual a 14 puntos (la calificación en Angola parte de 0 a 20 puntos, un estudiante es considerado aprobado con el mínimo de 10 puntos).
- INAGBE registra y selecciona los candidatos con ayuda de un sistema informático, los estudiantes seleccionados son sometidos a un examen, donde para ganar la beca deben alcanzar 14 puntos como mínimo. Si la cantidad de becas disponible sobrepasa la cantidad de candidatos, se establece una prioridad en orden descendente según la puntuación obtenida en el examen. Terminado el proceso de selección final, se realiza los trámites legales con el Ministerio de Relación Internacional del país destino y los estudiantes van para el mismo.
- INAGBE entra en contacto con SAE (representante de INAGBE en el país receptor) para que realice los trámites legales correspondientes a la recepción de los estudiantes en el país destino. SAE hace los trámites con el ministerio que se encarga de la educación superior en el país destino y con las universidades correspondientes, para que los estudiantes sean recibidos según la carrera otorgada. SAE recibe de INAGBE las documentaciones de los estudiantes para hacer los trámites legales necesarios, incluyendo la creación de las cuentas bancarias y el seguro de salud de los becados.
- Con la documentación y la confirmación de los estudiantes, el SAE hace el registro de los mismos en su sistema informático, las informaciones registradas son: los datos personales, la universidad en que se encuentra, la carrera, los datos bancarios, los datos del seguro de salud, los datos del estipendio anual y el resultado académico del año culminado.
- SAE informa al INAGBE la conducta de los estudiantes y el resultado académico del año culminado. Tanto los malos resultados académicos como las indisciplinas son factores suficientes para que el SAE o el INAGBE decidan dar de baja a un estudiante.

En el SAE Cuba, cuando un estudiante es dado de baja, sus datos son borrados de la base de datos para no sobrecargarla, también son borrados los datos de los estudiantes que terminan su formación, porque usan una base de datos hecha en Access que presenta limitaciones en la capacidad de almacenamiento. Con la eliminación de esta información sobre los estudiantes graduados o los que se fueron de baja, SAE no puede emitir un reporte verídico a INAGBE, y a su vez, INAGBE no podrá dar

un reporte confiable al gobierno de Angola, de cuántos estudiantes se formaron en el extranjero con éxitos y cuántos no, cuáles son los países con mejor resultado, las mayores causas de bajas académicas, entre otras informaciones estadísticas. La pérdida de la información histórica puede originar muchas más dificultades para la toma de decisiones oportunas.

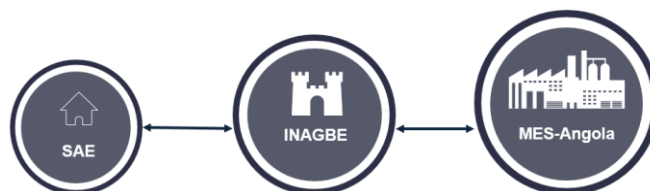


Ilustración 1. Flujo de información entre las instituciones.

1.3 Estudio de sistemas similares

El Ministerio de Educación superior de Angola en aras de tornar transparente el proceso de selección de los candidatos para becas de estudio, en el año 2013 decide desarrollar un sistema informático. Con la entrevista realizada se obtuvieron las siguientes informaciones relacionadas a las funcionalidades que este debía tener:

- 1- Procesamientos de candidaturas y atribución de becas (para el proceso de candidatura, se registran los estudiantes que cumplan con los requisitos del INAGBE. Para la atribución de becas, el listado pasa en un algoritmo que determina lo siguiente: Seleccionar los estudiantes cuya edad se acerca al 17 y que su promedio general se acerca al 20);
- 2- Pagos de los estudiantes y funcionarios del SAE;
- 3- Proceso de renovación de becas.

El sistema de gestión del INAGBE (SGI) se resume en los procesos antes descritos, que pueden ser referentes para la implementación de sistemas similares, como:

- Registrar estudiantes;
- Realización de renovación de becas;
- Búsqueda de estudiantes;
- Actualización de los datos de los estudiantes;

Sistema de Gestión Estudiantil (SGE):

Usando los métodos empíricos basado en observación y entrevistas al director del SAE-CUBA, se obtuvo toda la información referente al proceso de gestión estudiantil. Para facilitar dicho proceso, se cuenta con una base de datos creada en Access que realiza las siguientes operaciones:

- Añadir nuevo estudiante;
- Borrar el estudiante;
- Buscar el estudiante por nombre;
- Imprimir el formulario del estudiante;
- Centros de estudios (muestra un listado de las universidades del país);
- Resultados académicos (muestra el promedio general y si el estudiante promovió en el curso anterior);

El SGE posee algunos requisitos funcionales que pueden servir de referencia para otros sistemas similares; las funcionalidades de borrar estudiantes, buscar estudiante por nombre y resultados académicos generó la idea al autor, que en vez de borrar los datos del estudiante que fuera dado de baja académica o graduado por los problemas de almacenamiento, era necesaria la creación de un nuevo sistema informático que entre otras cosas, permitiera otras formas de búsquedas no limitadas solo por el nombre y almacenara los resultados académicos del estudiante de todos los años culminados, obteniendo una trayectoria académica.

Microsoft Access es un Sistema de gestión de bases de datos incluido en el paquete de programas de Microsoft Office. Algunas desventajas:

- Es una base de datos para pequeños escenarios.
- No es multiplataforma;
- Nivel de seguridad bajo;
- Su uso es inadecuado para proyectos de software con tiempo de respuesta críticos;

Con vista en lo planteado anteriormente el autor decide que esta filosofía de trabajo no es la más adecuada, si se trata de una institución que almacena grandes volúmenes de datos, es necesario una infraestructura que sea capaz de soportar. Es importante garantizar la seguridad y disponibilidad de la información.

Sistema de Gestión Universitaria:

Según Palmero y Remón, la UCI cuenta con un Sistema de Gestión Universitaria (SGU), desarrollado por el Centro de Informatización Universitaria, en el cual se gestiona toda la información referente a la formación de pregrado y postgrado. Cuenta con los siguientes módulos: Residencia, Pregrado, Egreso, Biblioteca, Ingreso, Extensión, Laboratorios, Producción, Investigación, Cooperación y Postgrado. Los módulos anteriores se controlan mediante un sistema de sesiones que garantizan el acceso eficiente a la información [11].

Debido a restricciones de acceso, a continuación, se describe el módulo Pregrado donde se encuentran cuatro funcionalidades disponibles al estudiante: Plan de estudio, tramites docentes, registro de asignatura y resumen de evaluaciones. En “tramites docentes” se permite imprimir las hojas de ratificación de matrículas; En la funcionalidad “resumen de evaluaciones” se visualiza la trayectoria académica del estudiante y una opción para imprimir los resultados obtenidos hasta el momento [12].

Con lo antes descrito se considera que el SGU comparte ventajas que pueden ser referencias para otros sistemas con características similares.

Para el análisis de los sistemas anteriormente descritos e identificar las particularidades del nuevo sistema que se desea implementar, a continuación, se muestran las características que se van a tener en cuenta:

- C1: Visualizar los estudiantes con baja académica; es una de las características imprescindible para la solución propuesta, pues ayuda al cliente a tener control de todos los estudiantes dados de baja académica.
- C2: Visualizar estudiantes graduados; una vez que los estudiantes se gradúan el cliente necesita tener fácil acceso y control a la información. La solución propuesta debe permitir visualizar información de los estudiantes graduados y su trayectoria académica.
- C3: Satisface las necesidades del SAE para mejor rendimiento; es una característica muy general que incluye, tener acceso a la trayectoria académica de los estudiantes, permitir hacer un listado con las cuentas bancarias y carnet de identidad de los estudiantes seleccionados para hacer el pago de estipendio, así como las políticas de acceso al sistema.
- C4: Reintegrar Estudiantes; esta característica debe permitir reintegrar a los estudiantes dados de baja académica, si por algún motivo al estudiante le sea permitido reincorporarse como becario.

- C5: Contribuye para el cumplimiento de los pilares de la seguridad informática; es imprescindible que la propuesta de la solución permita que solo los usuarios autorizados tengan acceso a la información, puedan modificar la información y que tenga disponible la información siempre que sea necesario.
- C6: Multiplataforma; actualmente es necesario que los sistemas puedan ser usados sin problemas, independientemente del sistema operativo.

Tabla 2. Representación de características convergentes entre los sistemas informáticos

Funcionalidades y Características / Sistemas de Gestión	C1	C2	C3	C4	C5	C6
SIG				X	X	X
SGU	X	X		X	X	X
SGE						

Como se observa en la tabla 2, de los sistemas informáticos identificados con el objetivo de gestionar la información de estudiantes, no se pudo seleccionar ninguno para solucionar el problema planteado, ya que no satisfacer la característica C3. Siendo para estos el problema de convergencia entre los sistemas, la solución propuesta se sujeta a características específicas de las debilidades identificadas en la base de datos implementada en el SAE. Es necesario un sistema enfocado en la docencia o en la gestión universitaria, pues habrá muchos módulos innecesarios para el cliente, se requiere un sistema que este enfocado principalmente a la gestión de los estudiantes.

Por otro lado, se concluye que el SIG y el SGU son sistemas que resuelven los problemas específicos para los que fueron implementados. El SGU tiene un enfoque mucho más amplio y enfocado para instituciones universitarias, mientras que el SIG es un sistema con enfoque en la selección de estudiantes que pretende ganar una beca de estudio en el extranjero o en el país Angola. Sin embargo, algunas de sus funcionalidades como la búsqueda por filtro, la posibilidad de extraer la información en PDF (en el caso de la propuesta de la solución se le permitirá extraer en Excel por comodidad del cliente) se puede tomar como referencia para la nueva implementación. Del sistema actualmente usado en el SAE (SGE) se aprovechará los datos almacenados. Además, se aprovechará el diseño de la interfaz que tiene para registro de estudiantes como

reseña para registrar estudiantes en la solución propuesta.

1.4 Metodología de Desarrollo de Software

Una metodología de desarrollo es la guía que contiene aquellos pasos y procedimientos que deben seguirse para desarrollar un software [13].

1.4.1 Selección de tipo de la Metodología

Las metodologías tradicionales son aquellas que ponen mayor énfasis en la planificación y control del proyecto y en la especificación precisa de requisitos y modelado. Las metodologías ágiles están más orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos pequeños, priorizan los recursos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso [14]. A continuación, se presenta una figura comparativa:

Metodologías Tradicionales	Metodologías Ágiles
Predictivos Orientado a procesos	Adaptativos Orientado a personas
Proceso rígido Se concibe como un proyecto	Proceso flexible Un proyecto es subdividido en varios proyectos más pequeños.
Poca comunicación con el cliente	Comunicación constante con el cliente.
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

Figura 1. Comparación de Metodología [15].

Según la comparativa hecha, se decide seleccionar la Metodología Ágil.

1.4.2 Comparación de Metodología Ágiles

AUP: en su sigla en inglés proceso unificado ágil (PUA) adopta una filosofía “en serie para lo grande” e “iterativa para lo pequeño” a fin de construir sistemas basados en computadora. Al adoptar las actividades en fase clásicas del PU -concepción, elaboración, contrición y transición-, el AUP brinda un revestimiento en serie que permite que el equipo visualice el flujo general del proceso de un proyecto de software, las

actividades son: modelado, implementación, pruebas, despliegue, configuración y administración del proyecto, administración del ambiente [16].

Scrum: es un método de desarrollo ágil de software concebido por Jeff Sutherland y su equipo de desarrollo. Las actividades estructurales que incorporan son: requerimientos, análisis, diseño, evolución y entrega. Dentro de cada actividad estructural, las tareas del trabajo ocurren con un patrón del proceso llamado sprint. El trabajo realizado dentro de un sprint (el número de estos que requiere cada actividad estructural varía en función de la complejidad y tamaño del producto) se adapta al problema en cuestión [16].

XP: es el enfoque más utilizado del desarrollo de software ágil. Aunque las primeras actividades con las ideas y los métodos asociados a XP ocurrieron al final de la década de 1980, el trabajo fundamental sobre la materia había sido escrito por Kent Beck. Una variante de XP es XP industrial se propuso en una época más reciente, las actividades son: planeación, diseño, codificación y prueba [16].

A continuación, se presenta la tabla comparativa, donde se ha hecho la selección de elementos comparativos en la tabla, según lo que es necesario para el sistema a ser desarrollado.

Características	SCRUM	AUP	XP
Documentación	Mucha	Regular o Mucha	Poca
Relación con el cliente	Poca	Regular	Constante
Uso en proyecto	Pequeño y Mediano	Pequeño y Mediano	Pequeño y Mediano
Enfoque	En las personas	En las actividades	En las personas
Refactorización del Código	No	No	Sí

Tabla 2. Comparación Ágil [16] [17].

Luego de analizar estas metodologías se escoge XP por las siguientes razones:

- Teniendo en cuenta las características del equipo de desarrollo, el cual está conformado por 1 individuo y la permanente interacción con el cliente, resulta apropiado la implementación de prácticas correspondientes a la metodología XP.

- El cliente necesita el sistema lo más rápido posible, porque le hacen falta funcionalidades importantes en su quehacer diario para cumplir con sus tareas. XP es apropiado porque permite en cada iteración implementar, hacer las pruebas y entregar una o varias historias de usuario, sabiendo que cada iteración tiene como máximo de tiempo 3 semanas.
- El cliente está disponible para formar parte del equipo de desarrollo para culminar el proyecto lo más rápido posible (aproximadamente 3 meses).
- Es abierta a los cambios y genera poca documentación lo que hace la entrega del software menos complicada y más satisfactoria tanto para los clientes como para el equipo de desarrollo.
- XP tiene un enfoque en la refactorización del código y además el equipo tiene dominio en las tecnologías seleccionadas.

1.5 Tecnologías, lenguaje de programación y herramientas

- **MERN Stack**

Según Bhandari es un entorno de desarrollo compuesto por 4 tecnologías independientes, Mongo DB, Express JS, React JS y Node JS, de ahí la abreviatura MERN [18].

MongoDB: Es una base de datos distribuida, basada en documentos y de uso general que ha sido diseñada para desarrolladores de aplicaciones modernas y para la era de la nube [19].

Mongo DB es una base de datos de código abierto y líder en bases de datos NoSQL. MongoDB está escrito en c++. Ofrece alta disponibilidad y escalabilidad a costa de consistencia y soporte transaccional. En términos prácticos, esto significa que, en lugar de tablas y filas, MongoDB utiliza documentos para hacerla flexible, escalable y rápida. Guarda estructuras de datos en documentos similares a JSON con un esquema dinámico (MongoDB utiliza una especificación llamada BSON para almacenar sus documentos), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida. Los documentos BSON sirven para mantener una lista ordenada de elementos, los mismos contienen tres componentes: Un nombre de campo, un tipo de datos y un valor. Estos documentos pueden tener esquemas diferentes, lo que significa que el esquema puede cambiar a medida que evoluciona la aplicación [20].

Express JS: Es un marco de Node.js es de código abierto muy ligero, extremadamente mínimo y flexible diseñado para crear aplicaciones web, utilizando API. Express se ejecuta como un módulo o un elemento dentro del entorno Node.js. A diferencia de otros marcos, es mucho más minimalista y ofrece características

avanzadas. Al trabajar como una capa sobre Node.js, puede ayudar a diseñar aplicaciones de una sola página, múltiples páginas e híbridas. Proporciona seguridad, líneas de códigos reducidas, por lo tanto, un desarrollo mucho más rápido y fácil accesibilidad a complementos de terceros lo convierte en el marco Node.js más popular entre los desarrolladores [18].

React JS: Una biblioteca de JavaScript para construir interfaz de usuario [21]. Es basada en componentes siendo de código abierto, es una de las más utilizadas para crear interfaz de usuario del lado del cliente de forma natural para aplicaciones de una sola página. Primero fue desarrollado por el ingeniero Jordan Walke de Facebook y se actualiza constantemente por la misma empresa y la comunidad de desarrolladores de todo el mundo. Se implementó por primera vez en la fuente de noticias de Facebook en 2011 [18].

Node Js: Es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome [22]. Es el entorno de servidor web de código abierto más popular de la actualidad desarrollado por Ryan Dahl y Joyent Inc (actualmente propiedad de Samsung Electronics). Al ser un entorno de tiempo de ejecución, Node.js ayuda a ejecutar códigos JavaScript fuera de un navegador. Utiliza el motor de Google chrome para ejecutar JavaScript. En términos simples, ayuda a traducir los códigos de JavaScript al lenguaje de nivel de máquina, por lo tanto, una máquina puede entender los códigos de JavaScript sin la necesidad de un navegador. Este factor es clave para desarrollar aplicaciones full-stack. Node.js en su conjunto facilita mucho el desarrollo de aplicaciones multiplataforma, del lado del servidor y de red [18].

- **MEAN Stack**

Es un conjunto de software usado para el desarrollo de aplicación web de una sola página. La abreviatura MEAN (MongoDB, Express, Angular, Node.js) fue originalmente acuñada por Valeri Karpov en el año 2013 [23].

Angular JS: Es un marco de creación de aplicaciones web de una sola página del lado del cliente que ofrece una variedad de herramientas que simplifican y aceleran la creación de contenido web dinámico y de aplicaciones de una sola página. Al usar angular, el desarrollador puede evitar escribir la mayor parte del código repetitivo que se incluiría en una aplicación web, como la manipulación directa de DOM. Angular ofrece herramientas como el enlace de datos, el enrutamiento de vistas, la vista de árboles, la inyección de dependencias y el enrutamiento [24].

- **MEVN Stack**

Este marco consta de una base de datos MongoDB, Express, Vue.js y Node.JS. Toda la aplicación se ejecuta en código JavaScript, con los datos de la aplicación almacenados en esquemas similares a JSON en una base de datos MongoDB [25].

Vue.js: es un marco progresivo para construir interfaz de usuario desarrollado por Evan You. Así como sus homólogos (React y Angular) es de código abierto, modificable y mejorable. Los Vue-models son objetos JavaScript simples, lo que hace que la interacción con el estado de la aplicación sea sencilla. Cuando la acción desencadena una modificación en el estado de la aplicación, la vista se actualiza automáticamente [25].

Para la selección de una de las tres tecnologías antes mencionadas, es importante resaltar que solo la tecnología de construcción de interfaz del usuario deberá cambiar. Por lo tanto, el criterio de selección se basaría en los tres competidores de la actualidad (Angular JS, React JS y Vue.JS).

Tabla 3. Comparación entre las tres tecnologías

Tecnología	Características				
	Curva de aprendizaje	Flexibilidad	Tipo	Lenguaje	Rendimiento
Angular JS	Alta	Baja	Framework	TypeScript	Alto
React JS	Media	Alta	Librería	JSX /Javascript	Alto/Medio
Vue.JS	Baja	Alta	Framework	Html /Javascript	Medio/Bajo

La información que se muestra en la tabla 3 fue retirada de Robledo y Petukhova [26]. Una encuesta realizada por *Sitecore JavaScript Services* en la comunidad de desarrolladores donde las respuestas que se buscaban eran las siguientes:

- He usado y volvería a usar;
- He usado y no volvería a usar;
- He oído hablar y me gustaría aprender;
- He oído hablar y no me interesa;
- No he oído hablar.

En la ilustración 2 se muestra el resultado probabilístico de dicha encuesta.



Figura 2. Resultado probabilístico de la encuesta comparativa [27].

Para un estudio más profundo se ilustra la tendencia de búsqueda en Google de los últimos años.



Figura 3. Tendencia de búsqueda y porcentajes en Google Trends [27].

Analizando lo antes descrito, el autor decide crear un entorno de desarrollo que tenga las siguientes características:

- Flexible (que sea posible la estructuración del proyecto según la filosofía de trabajo);
- Curva de aprendizaje favorable;
- El código fuente lo más organizado posible (sin tener que usar HTML de forma obligatoria, pues con la existencia de tags sin cerrar, ocasiona inestabilidad en la interfaz);
- Un alto rendimiento;
- Que sea popular (fácil de encontrar referencias bibliográficas).

Con las características planteadas se decide **React JS** como librería de interfaz de usuario, se asume la tecnología **MERN Stack**, por agregar React JS en su entorno de trabajo y ser ampliamente conocida por el programador (XP señala que la tecnología a utilizar debe ser dominada por el equipo de desarrollo).

Lenguaje de programación

- **JavaScript**

Es un lenguaje interpretado usado múltiples propósitos, pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo que vemos JavaScript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores más exitosos fue transformar el código JavaScript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejorada capacidad permitió superar antiguas limitaciones de rendimiento y confirmar el lenguaje JavaScript como la mejor opción para la web [28].

Se hará uso de JavaScript para el desarrollo de la aplicación por ser la base del entorno planteado por el autor.

1.6 Herramientas a utilizar

- **NoSQLBooster 5.1.14**

Es un software para trabajar en MongoDB con interfaz gráfica que ofrece una verdadera experiencia de sentido inteligente que brinda sugerencias mientras van escribiendo. Tiene un servicio incorporado de lenguaje que conoce todas las terminaciones, métodos, propiedades, variables, palabras clave posibles, incluso los nombres de colección MongoDB, nombres de campo y operadores.

Con NoSQLBooster para MongoDB, se puede ejecutar consultas SQL SELECT. El soporte SQL incluye SQL JOINS, funciones, expresiones, agregación para colecciones con objetos anidados y matrices.

Schema Analyzer es una herramienta práctica incorporada. Debido a la función sin esquema, las colecciones en MongoDB no tienen un documento de esquema para describir el tipo de datos del campo, la estructura de la colección y las validaciones. Con Schema Analyzer Tool, es posible tener un documento en que se puede escribir los esquemas de la colección, determinar tipo de datos del campo [29].

- **Visual Studio Code 1.39.2**

Es un editor de código fuente ligera pero potente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity). Posee un sistema inteligente de sugerencias y complementaciones para variables, métodos y módulos importados, además de tener herramientas para la terminal de depuración de errores, cuenta con el soporte SCM dentro de su editor, integración de Git [30].

- **Visual Paradigm 15.1**

Visual Paradigm es una herramienta para visualizar y diseñar elementos de software, para ello utiliza UML (UML 2.1) y ofrece una gama de facilidades para el modelado de aplicaciones. Está orientada a la creación de diseños usando el paradigma de programación orientada a objetos [11].

- **Owasp zap 2.7.0**

Owasp ZAP de la fundación Owasp que trabaja para mejorar la seguridad del software a través de sus proyectos de software de código abierto liderados por la comunidad, cientos de capítulos en todo el mundo, decenas de miles de miembros y organizando conferencias locales y globales [31]. Owasp zap es una herramienta usada en el testeado de penetración en sistemas informáticos, con un gran enfoque en sistemas web.

Con vista en lo planteado el autor declara el listado de las herramientas a usar para el desarrollo de la solución propuesta:

- **Entorno de desarrollo:** Visual Studio Code 1.39.2.
- **Herramienta de Modelado:** Visual Paradigm 15.1
- **Gestor de base de datos:** NoSQLBooster 5.1.14.
- **Marco de trabajo:** MERN Stack.
- **Herramienta de Prueba:** Owasp ZAP 2.7.0.

1.7 Conclusiones del capítulo

En el capítulo fueron explicados los principales resultados obtenidos una vez realizadas las tareas 1, 2 y 3 de la investigación, a partir de lo cual se concluye que:

- El análisis del sistema utilizado el SAE demostró la necesidad de realizar una nueva herramienta, debido a que las limitaciones presentes no suplen las necesidades hoy existentes. Además, el

estudio de los sistemas similares permitió identificar características útiles a tener en cuenta en la propuesta de la solución.

- Fue seleccionada la metodología XP, debido a que cumplía con los requisitos para la solución propuesta, teniendo en cuenta el ambiente de desarrollo creado para la misma. Además, hace posible en cada iteración una entrega de software funcional según las necesidades del cliente.
- La selección del marco de trabajo MERN (MongoDB, Express, React JS y Node JS) permitirá la creación de un sistema informático web de una sola página, logrando que el sistema informático web no haga que el navegador se recargue y que sean más rápidas las peticiones al servidor.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

En este capítulo se presenta la propuesta del Sistema de Gestión de Estudiantes, con el cual se puede gestionar informaciones relacionados a los estudiantes angoleños que se encuentran en Cuba. Se muestran artefactos propios de la metodología XP en la fase planeación y diseño, tales como: las Historias de Usuario con las que se guía el proceso de desarrollo, las tarjetas CRC como metodología para el diseño del paradigma orientado a objetos [32]. Por otra parte, se hace referencia a los usuarios del sistema, el modelo de datos y también se muestra la lista de reserva del producto e iteraciones para la producción de entregables del sistema.

2.1 Descripción de la propuesta de la Solución

Para la solución a la problemática antes planteadas (ver figura 3), la propuesta de solución consta de 2 módulos. En el módulo Estudiantes se tendrán en cuenta las funcionalidades que permite a los usuarios del sistema, adicionar, ver, buscar, transitar de año, dar de baja, graduar, adicionar resultados académicos a un estudiante. Mientras que el módulo Configuración será el encargado de gestionar provincias, universidades, carreras, áreas de formación y usuarios del sistema.

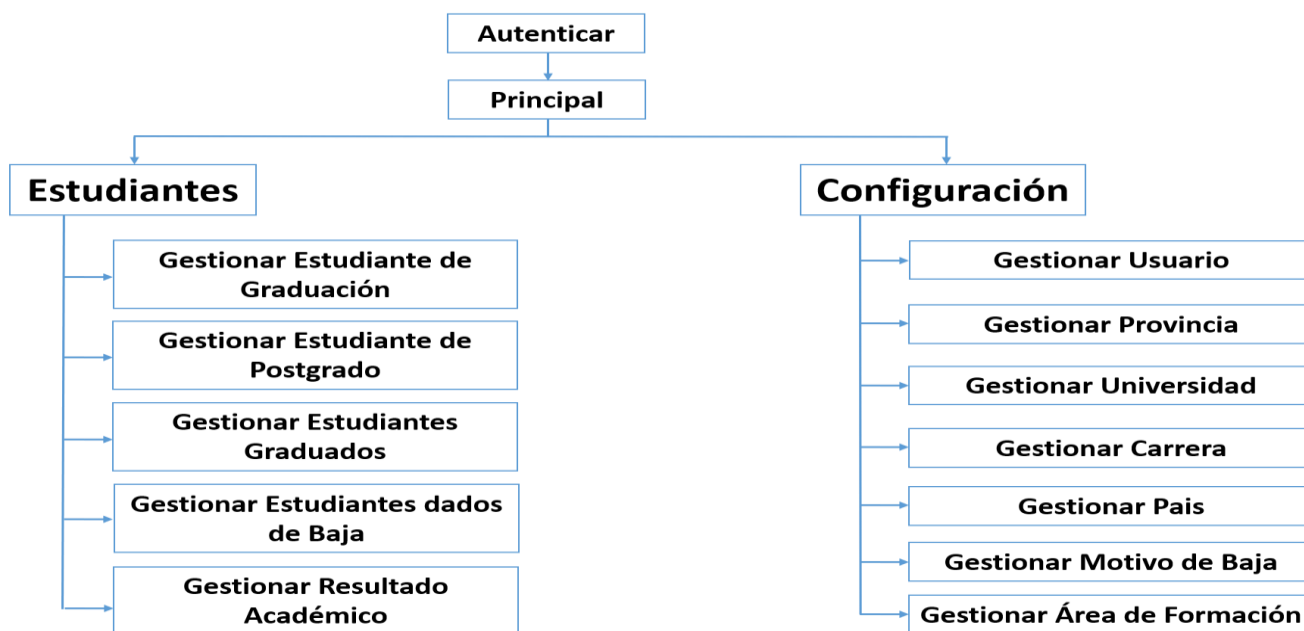


Figura 4. Propuesta de la solución.

2.2 Usuarios del Sistema

Los usuarios podrán interactuar con el sistema según sus privilegios establecidos en niveles, donde:

Nivel 0: Los usuarios de nivel 0 son los que tienen acceso a todas las funcionalidades del sistema. Ejemplo de usuario de nivel 0 es el director del SAE y la administradora financiera del SAE.

Nivel 1: Los usuarios de nivel 1 son los que no tienen permitidos usar las funcionalidades de dar de baja a un estudiante y graduar un estudiante. Además, no tienen permiso para añadir, eliminar, ver, buscar o editar usuario, solo les es permitido editar su información: nombre, email, usuario, foto y contraseña. Teniendo en cuenta las restricciones, tienen permiso para las demás opciones.

2.3 Planificación

La planeación es una de las fases de la metodología XP, la cual se describe en la siguiente sección. Durante la planificación se priorizan las Historias de Usuarios (HU) y se acuerda el alcance de la entrega. Esta fase consiste en una o varias reuniones grupales de planificación y el resultado es un plan de entregas. Se realiza una estimación del esfuerzo que costará implementar cada historia de usuario utilizando como medida el punto (una semana ideal de trabajo, el máximo de punto para cada Historias de Usuarios es 3) [33].

2.3.1 Lista de reserva del producto

Con motivo de establecer una organización a la hora de desarrollar un proyecto de software se realiza la lista de reserva de productos. Este artefacto, típico de la metodología XP, constituye una opción para dar prioridad a las funcionalidades a implementar en un proyecto. Es por ello por lo que se muestra a continuación la lista de reserva del producto confeccionada para el desarrollo de las funcionalidades presentes en la propuesta de solución.

Tabla 4. Lista de reserva del producto

Ítem	Descripción
Prioridad: Alta	
1	Adicionar Estudiante
2	Listar Estudiantes
3	Ver Estudiante

4	Dar de Baja a un Estudiante
5	Buscar por Nombre
Prioridad: Media	
6	Graduar Estudiante
7	Transitar de Clase a un Estudiante
8	Editar Estudiante
9	Listar Estudiantes dados de Baja
10	Listar Estudiantes Graduados
11	Reintegrar Estudiante dado de Baja
12	Buscar por Filtros
13	Gestionar Usuario
Prioridad: Baja	
14	Gestionar Resultado Académico
15	Gestionar País
16	Gestionar Provincia
17	Gestionar Universidad
18	Gestionar Área de Formación
19	Gestionar Carrera
20	Gestionar Motivo de Baja

2.3.2 Historias de Usuario

Las Historias de Usuario (HU) son el artefacto eje de la metodología XP debido a que estas describen las funcionalidades que comprenderá el sistema. Las mismas son escritas por el cliente empleando un lenguaje transparente y conciso que deje bien claro las necesidades a suplir con el sistema [34]. Permiten darle respuesta inmediatamente a las funcionalidades cambiantes en el transcurso del desarrollo y sirven de base

para las pruebas de funcionamiento del sistema. También se emplean para estimar el tiempo y para la realización de los planes de entregas [35]. Las HU están divididas por iteraciones:

Iteración 1: Se plantean las principales funcionalidades relacionadas con la gestión de los estudiantes y el resultado académico.

Iteración 2: Se identifican las funcionalidades relacionadas a la gestión de las carreras, provincias, universidades y área de formación.

Iteración 3: Se les da tratamiento a las búsquedas por filtros, a la gestión de motivos de baja y país.

Iteración 4: Se le da tratamiento a la gestión de usuarios teniendo en cuenta los permisos y autenticación en el sistema.

A continuación, se muestran algunas de las HU realizadas durante la fase de planificación, que abarcan las funcionalidades correspondientes en el sistema propuesto. Es importante apuntar que las demás HU se encuentran en el anexo 1.

Tabla 5. HU-1 Adicionar Estudiante

Historia de Usuario 1	
Nombre: Adicionar Estudiante	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
Puntos Estimados: 3	Puntos Reales: 3
Descripción: El usuario debe ir al módulo Estudiantes y escoger entre estudiante de postgrado o licenciatura, después podrá presionar en el botón adicionar estudiante.	
Observación: Todos los usuarios tienen permisos para adicionar estudiante. Para adicionar un estudiante el usuario debe verificar si la provincia, carrera o universidad del estudiante ya se encuentra adicionado en el sistema.	

Tabla 7. HU-3 Dar de Baja

Historia de Usuario 4	
Nombre: Dar de baja	Iteración Asignada: 1

Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
Puntos Estimados: 1	Puntos Reales: 1
<p>Descripción: El usuario debe ir al módulo Estudiantes y escoger entre estudiante de postgrado o licenciatura, después en el listado de estudiantes debe seleccionar el estudiante y seleccionar el botón visualizar (botón que se torna visible después que se haga la selección de uno o más estudiantes), después puede seleccionar el botón en dar baja. A los usuarios de nivel 1 no les será visible el botón Dar de Baja.</p>	
<p>Observación: Los usuarios con privilegio para dar de baja a un estudiante son los de nivel 0. Es necesario filtrar el estudiante deseado en la lista de estudiantes en el caso de no estar visible de antemano.</p>	

Tabla 21. HU-17 Gestionar Universidad

Historia de Usuario 17	
Nombre: Gestionar Universidad	Iteración Asignada: 2
Prioridad en Negocio: Baja	Riesgos de Desarrollo: Baja
Puntos Estimados: 2	Puntos Reales: 2
<p>Descripción: El usuario debe ir al módulo configuración y escoger universidad, luego se listan los datos en una tabla.</p> <p>Puede adicionar, editar o eliminar universidad. Para eliminar una universidad debe tener seleccionada al menos una para tener el botón eliminar disponible. Para que el botón editar aparezca debe seleccionar solo una universidad.</p>	
<p>Observación: Para adicionar una universidad debe verificar que la provincia de la universidad ya se encuentra adicionado en el sistema.</p>	

Tabla 16. HU-12 Buscar por Filtros

Historia de Usuario 12

Nombre: Buscar por Filtros	Iteración Asignada: 3
Prioridad en Negocio: Media	Riesgos de Desarrollo: Media
Puntos Estimados: 5	Puntos Reales: 5
<p>Descripción: El usuario debe ir al módulo Estudiantes y podrá escoger entre las siguientes opciones: estudiante de postgrado, graduados, verificar baja o de licenciatura. Luego se listan los datos de los estudiantes en una tabla, bajo la caja de texto, se debe seleccionar el botón filtrar porque todos los filtros se encuentran ocultos. Se puede hacer todas las combinaciones posibles para que la búsqueda sea más eficiente, también la caja de texto funcionará para combinar la búsqueda por nombre con los demás filtros. Se puede filtrar por: provincia, universidad, carrera, área de formación, sexo, año académico</p>	
<p>Observación: Los filtros son seleccionados, al presionar en cada filtro, también puede escribir en los filtros para buscar o que se desea.</p>	

Tabla 24. HU-20 Gestionar Motivo de Baja

Historia de Usuario 20	
Nombre: Gestionar Motivo de Baja	Iteración Asignada: 3
Prioridad en Negocio: Baja	Riesgos de Desarrollo: Baja
Puntos Estimados: 2	Puntos Reales: 1
<p>Descripción: El usuario debe ir al módulo configuración y escoger Motivo de Baja, luego se listan los datos en una tabla: el motivo de baja, la descripción.</p> <p>Puede adicionar, editar o eliminar nuevo motivo de baja. Para eliminar motivo de baja debe tener seleccionada al menos una para tener el botón eliminar disponible. Para que el botón editar aparezca debe seleccionar solo un motivo de baja.</p>	
<p>Observación:</p>	

Tabla 17. HU-13 Gestionar Usuario

Historia de Usuario 13	
Nombre: Gestionar Usuario	Iteración Asignada: 4
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Puntos Estimados: 6	Puntos Reales: 7
<p>Descripción: El usuario debe ir al módulo configuración y escoger usuario, luego se listan los datos de usuarios en una tabla. Se podrá ver la foto, nombre, usuario, email y el nivel de acceso.</p> <p>Puede adicionar, editar o eliminar nuevo usuario. Para eliminar usuario debe tener seleccionada al menos una para tener el botón eliminar disponible. Para que el botón editar aparezca debe seleccionar solo un usuario.</p> <p>Solo los usuarios de nivel 0 tendrá acceso al apartado usuario. Para editar el usuario solo podrá cambiar el nivel de acceso, los demás datos no.</p> <p>Los datos, como contraseña, nombre, email, usuario e imagen, solo pueden ser cambiados por el propietario de la misma cuenta en la cabecera del sistema, seleccionar en el nombre de usuario e en seguida en perfil.</p> <p>Para autenticarse es necesario en la página login, entrar con el nombre de usuario y la contraseña. La contraseña entrada es comparada con el hash en la base de datos, comprobando el usuario y la contraseña, si son correctas tendrá acceso al sistema, según el nivel que posea.</p> <p>Observación: Al adicionar nuevo usuario, puede seleccionar una imagen o no, debe poner una contraseña por defecto y después avisar al usuario para cambiar la contraseña una vez este autenticado en el sistema. Al seleccionar el nivel de acceso solo tendrá 2 opciones: 0 tiene acceso total al sistema, 1 acceso limitado.</p>	

2.3.3 Requisitos no funcionales

Los requisitos no funcionales son propiedades que hacen al producto atractivo, usable, rápido o confiable. Se refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar. Se conocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el software [36].

Portabilidad:

- **RNF1:** Para el despliegue del software, deberá estar instalado en el servidor el sistema operativo Linux o Windows, donde se encuentre instalado Node.js versión 15 o superior, Apache 2.4.9.
- **RNF2:** La base de datos que utilizará el sistema como medio de almacenamiento de la información estará soportada sobre un gestor de bases de datos MongoDB.

Usabilidad:

- **RNF3:** Las vistas del sistema deben indicar en cada momento la acción que se está realizando, así como los íconos deben estar representados por una imagen acorde a la acción que se está realizando mediante el mismo.

Seguridad:

- **RNF4:** El sistema de autenticación registrará datos de los usuarios y tendrá control de la identidad de los mismos.
- **RNF5:** Debe proveer algún mecanismo seguro de encriptación y transferencia de datos.
- **RNF6:** Se debe mantener una seguridad a nivel de usuarios y contraseñas codificadas para el acceso a la base de datos.

Eficiencia:

- **RNF7:** El sistema debe ser capaz de responder satisfactoriamente ante la concurrencia de los usuarios que interactúen con el mismo.

2.3.4 Estimación de Esfuerzo por Historias de Usuario

Con vistas al correcto desarrollo del sistema propuesto, se realizó una estimación por HU identificadas y definidas anteriormente, se hace para saber cuánto debe durar el proyecto en tiempo. Teniendo en cuenta que 1 punto de estimación equivale a 1 semana y que las HU tienen como máximo 3 puntos para su implementación, según la metodología XP, se muestra la estimación en días en la tabla 25 (la estimación debe estar entre 1 día a 21 días, para que no sea mayor que 3 puntos).

Como consecuencia de este proceso se obtuvo el siguiente resultado:

Tabla 25. HU-20 Estimación de Esfuerzo por HU

No.	Historia de Usuario	Estimación (días)
1	Adicionar Estudiante	3
2	Lista Estudiantes	1
3	Ver Estudiante	1
4	Dar de Baja a un Estudiante	1
5	Buscar por Nombre	1
6	Graduar Estudiante	1
7	Transitar de Clase a un Estudiante	1
8	Editar Estudiante	2
9	Listar Estudiantes dados de Baja	1
10	Listar Estudiantes Graduados	1
11	Reintegrar Estudiante dado de Baja	1
12	Buscar por Filtros	5
13	Gestionar Usuario	6
14	Gestionar Resultado Académico	2
15	Gestionar País	2
16	Gestionar Provincia	2
17	Gestionar Universidad	2
18	Gestionar Área de Formación	2
19	Gestionar Carrera	2
20	Gestionar Motivo de Baja	2

2.3.5 Plan de Iteraciones para entregables

El plan de entrega está compuesto por iteraciones de no más de 4 semanas. En la primera iteración se definen todos los elementos arquitectónicos y técnicos para dar estructura al sistema, la cual pueda ser utilizada durante el transcurso de desarrollo del proyecto [35]. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción [33].

El plan de entregas se basa en las Historias de Usuario. Una vez identificadas las mismas, se definieron niveles de prioridad, lo que posibilitó dividir el desarrollo en 4 iteraciones de acuerdo con la prioridad de cada HU y la relación en cuanto a funcionalidad entre ellas.

Iteración 1: Se plantea las principales funcionalidades relacionadas con la gestión de estudiantes y resultado académico.

Iteración 2: Se identifican las funcionalidades relacionadas con la gestión de las carreras, provincias, universidades y área de formación.

Iteración 3: Se le da tratamiento a la búsqueda por filtros, a la gestión de motivos de baja y país.

Iteración 4: Se le da tratamiento a la gestión de usuarios teniendo en cuenta los permisos y la autenticación en el sistema.

Tabla 26. HU-20 Estimación de Esfuerzo por HU

Iteración	HU	Estimación (día)	Semanas
1	Adicionar Estudiante	3	3
	Lista Estudiantes	1	
	Ver Estudiante	1	
	Dar de Baja a un Estudiante	1	
	Buscar por Nombre	1	
	Graduar Estudiante	1	

	Transitar de Clase a un Estudiante	1	
	Editar Estudiante	2	
	Listar Estudiantes dados de Baja	1	
	Listar Estudiantes Graduados	1	
	Reintegrar Estudiante dado de Baja	1	
	Gestionar Resultado Académico	2	
2	Gestionar Provincia	2	3
	Gestionar Universidad	2	
	Gestionar Área de Formación	2	
	Gestionar Carrera	2	
3	Gestionar País	2	3
	Buscar por Filtros	5	
	Gestionar Motivo de Baja	2	
4	Gestionar Usuario	6	3
Total	12 semanas = 3 meses		

2.4 Diseño del Sistema

2.4.1 Tarjetas CRC

Las tarjetas CRC (clases, responsabilidades y colaboración) son una representación de las entidades del sistema a las cuales se les asignan responsabilidades y colaboraciones. El formato físico de las tarjetas CRC facilita la interacción entre clientes y el equipo de desarrollo, en sesiones en las que se aplican técnicas de grupos como tormenta de ideas o juego de roles, y se ejecutan escenarios a partir de especificación de requisitos o historias de usuarios [32]. De esta forma, van surgiendo las entidades del sistema junto con sus responsabilidades y colaboraciones. Luego, en un estado de diseño avanzado o ya en la implementación del sistema, las tarjetas CRC se convierten en clases con métodos, atributos, relaciones de herencia, composición o dependencia.

Durante el diseño de la aplicación se elaboraron un total de 11 tarjetas CRC, de las que se exponen a continuación 3 de ellas, las restantes se puede ver en el anexo 2:

Tabla 27. Tarjeta CRC Estudiante

Tarjeta CRC	
Clase: Estudiante	
Responsabilidad	Colaboración
<p>Contiene toda la información referente a un estudiante.</p> <p>Permite adicionar un estudiante, cambiar los atributos de un estudiante.</p> <p>Permite filtrar o buscar</p>	<p>País</p> <p>Provincia</p> <p>Universidad</p> <p>Área de formación</p> <p>Carrera</p> <p>Usuario</p>

Tabla 28. Tarjeta CRC Graduado

Tarjeta CRC	
Clase: Graduado	
Responsabilidad	Colaboración
<p>Contiene toda la información de un estudiante graduado.</p> <p>Permite adicionar un estudiante graduado, cambiar los atributos de un estudiante.</p> <p>Permite filtrar o buscar</p>	<p>Estudiante</p>

Tabla 29. Tarjeta CRC Baja

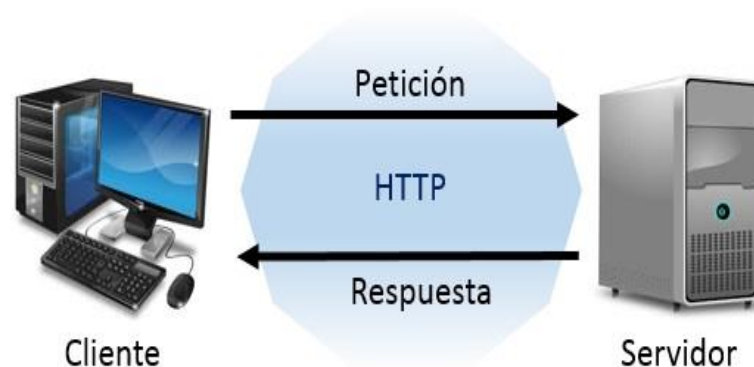
Tarjeta CRC

Clase: Baja	
Responsabilidad	Colaboración
Contiene toda la información de un estudiante dado por baja. Permite adicionar un estudiante dado por baja, cambiar los atributos de un estudiante. Permite filtrar o buscar	Estudiante Motivo de Baja

2.4.2 Arquitectura

El estándar *IEEE 1471*¹ define la arquitectura de software como “la organización fundamental de un sistema incorporados en sus componentes, sus relaciones entre sí y con el medio ambiente, y los principios que guían su diseño y evolución”. La arquitectura del software es importante porque afecta el rendimiento, la solidez, la capacidad de distribución y la capacidad de mantenimiento de un sistema.

En una arquitectura Cliente-Servidor, la funcionalidad del sistema se organiza en servicios, con cada servicio entregado desde un servidor separado. Un cliente realiza una solicitud a un servidor y espera hasta que recibe una respuesta. Los clientes acceden a los servicios proporcionados por un servidor a través de llamadas a procedimientos remotos, usando un protocolo de solicitud-respuesta como el protocolo HTTP usado en la Internet.



¹IEEE Standard 1471: Práctica recomendada para la descripción arquitectónica de sistemas de software intensivo. Desarrollado por la IEEE Computer Society.

Figura 4. Arquitectura cliente-servidor

2.4.2.1 Patrón Arquitectónico

Se decide determinar el patrón arquitectónico Modelo-Vista-Controlador. Este patrón separa la lógica de negocio en tres clases diferentes:

Modelo: administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

Vista: maneja la visualización de la información.

Controlador: controla el flujo de datos entre la vista y el modelo. Entre las ventajas aprovechadas de este patrón para el desarrollo de la solución se encuentran las siguientes:

- **Soporte de múltiples vistas:** dado que la vista se haya separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas del sistema informático web pueden utilizar el mismo modelo de objetos mostrado de maneras diferentes.
- **Adaptación al cambio:** los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación. Con la utilización de este estilo, el modelo no depende de las vistas y agregar nuevas opciones de presentación generalmente no genera afectaciones.

La elección de este patrón está basada en el hecho de que convierte a la aplicación en un paquete modular y de desarrollo rápido permitiendo que algunos aspectos de la estructura del sistema puedan cambiar independientemente de otros [11].

La figura 3 muestra una imagen conceptual del funcionamiento de MERN Stack donde React JS se comunica haciendo solicitudes al Node JS utilizando el marco Express. Node JS conectará el servidor back-end con MongoDB que se usa como base de datos y luego responderá en forma JSON al front-end.

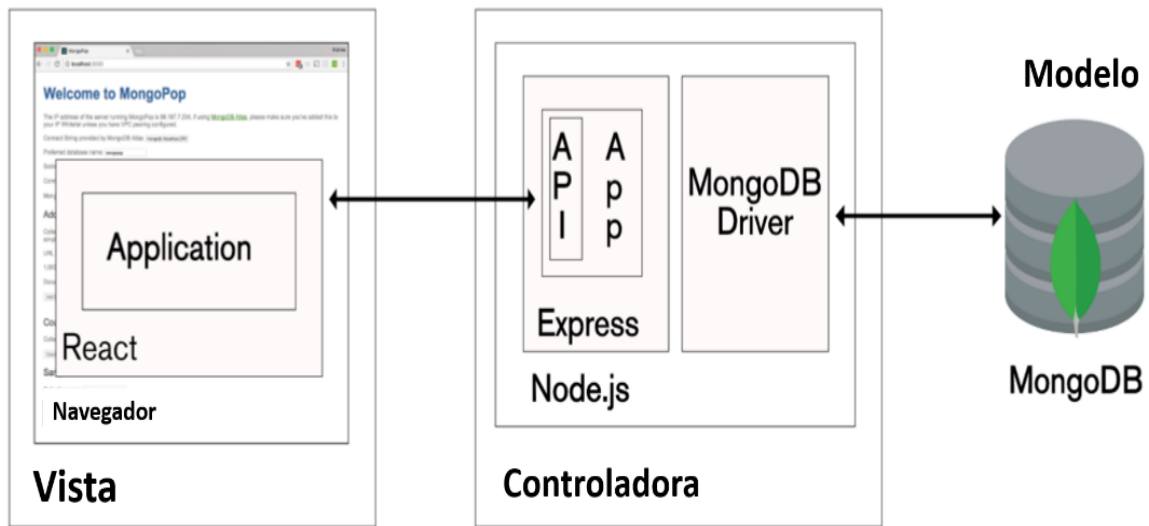


Figura 5. Diseño conceptual de la arquitectura [18].

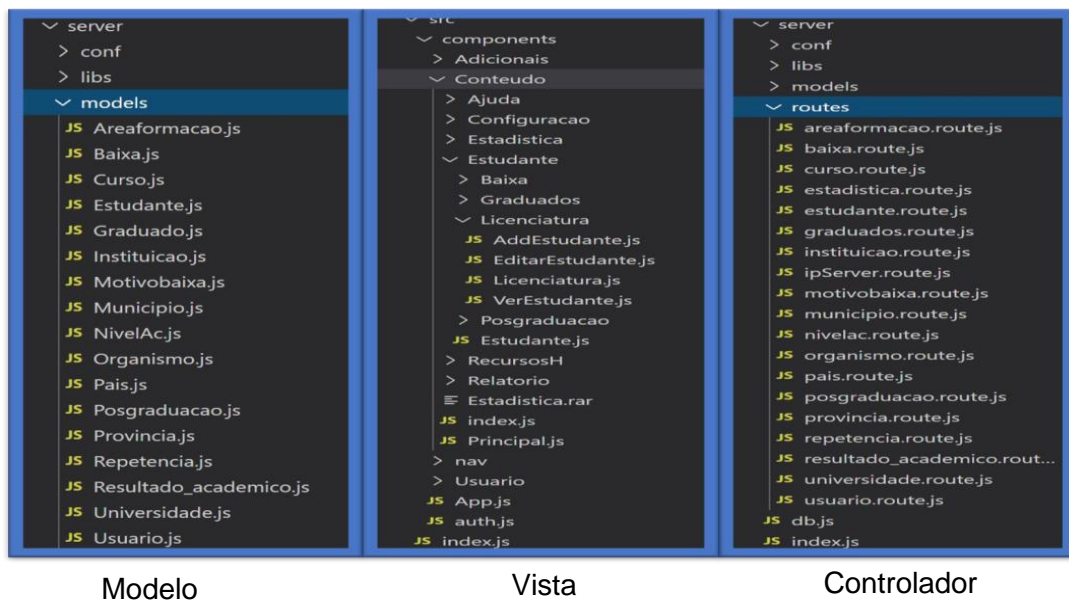


Figura 6. Evidencia de la arquitectura.

2.4.3 Patrones de Diseño

Los patrones de diseño brindan una solución probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares [37]. Un patrón es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Estos patrones identifican Clases, Instancias, Roles, Colaboraciones y la distribución de responsabilidades.

2.4.3.1 Patrones GoF (Gang of Four)

En el año 1994, apareció el libro “Design Patterns: Elements of Reusable Object Oriented Software” escrito por los ahora famosos Gang of Four (Pandilla de los cuatro) integrada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Estos recopilaron y documentaron 23 patrones de diseño aplicados usualmente por expertos diseñadores de software orientado a objetos [38].

Para el desarrollo de aplicaciones web, se identificaron 8 patrones de diseño Gof de los 23 en los procesos de desarrollo software [39]. A continuación se muestran los que fueron usados en el presente trabajo:

Creacionales:

Constructora (Builder): un patrón de creación de objetos delega la instanciación en otro objeto [41].

```
1  const express = require('express');
2  const router = express.Router();
3  const multer = require('multer');
4  const path = require('path');
5
6
7  const Estudiante = require('../models/Estudiante');
8
9  router.get('/', async (req,res) => {
10     const estudiantes = await Estudiante.find();
11     res.json(estudiantes);
12 });
13
```

Figura 7. Evidencia del patrón constructora.

Estructurales:

Decorador (Decorator): Añade funcionalidad a una clase dinámicamente [39]. En el documento se evidencia haciendo uso de la clase Api.js.

Adaptador (Adapter): utiliza la herencia para componer interfaz o implementaciones [40].

En el trabajo se puede evidenciar el cumplimiento de estos patrones en el empleo del index.html y en los componentes Menuleft.js, Head.js, Footer.js y Sumenu.js que son usados en diversas partes del sistema.

Comportamiento:

Iterador (iterator): se utilizó para acceder a los elementos de un objeto para su representación en las vistas y funcionalidades; en las clases controladoras, necesarias para acceder a la información contenida en cada instancia del objeto dentro de la colección del tipo de objeto referido [42].

```
338     });
339     let ident = '';
340     for (let i = 0; i < this.state.provinciaApi.length; i++) {
341         console.log(this.state.provinciaApi[i].nome)
342         if (this.state.provinciaApi[i].nome === e.target.value) {
343             ident = this.state.provinciaApi[i]._id;
344             console.log(this.state.provinciaApi[i]._id)
345         }
346     }
347     console.log(ident)
348     BuscarUniversidadProv(ident, 'universidade')
349     .then(data => {
350         this.setState({ universidadeApi: data })
351         console.log(data)
352     })
353 }
354 }
355 }
356 }
```

Figura 8. Evidencia del patrón iterador.

2.5 Conclusiones del capítulo

En el capítulo se expuso los principales resultados obtenidos una vez realizada la tarea 4 de la investigación, a partir de lo cual se concluye que:

- La división del sistema en 2 módulos contribuyó a un diseño reutilizable y sencillo de entender e implementar.
- La identificación de los 7 requisitos no funcionales y de las 20 historias de usuario en la fase de planeación de la metodología permitió determinar cuáles son las partes del software a ser implementado con prioridad. Además, la estimación del tiempo identificó que el proyecto era entregable en el tiempo adecuado según XP y según el fondo de tiempo para un trabajo de este tipo.
- La elaboración de 11 tarjetas CRC facilitó la detección temprana de errores en el diseño del sistema.
- El uso de los patrones MVC y GoF proporcionó un criterio objetivo para asignar responsabilidades a las clases y definir las relaciones que se establecen entre ellas, permitiendo un diseño flexible y fácil de mantener.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

En el presente capítulo se analizan los aspectos relacionados con la implementación y las pruebas realizadas a la aplicación. Se exponen los elementos fundamentales del modelo de datos, se describen los métodos y técnicas para la realización de las pruebas con el propósito de validar la propuesta de solución.

3.1 Desarrollo

3.1.1 Modelo de Datos

Un modelo es un conjunto de herramientas conceptuales para describir datos, sus relaciones, su significado y sus restricciones de consistencia [44].

MongoDB es un sistema de base datos potente, flexible, de alto rendimiento, de expansión fácil de almacenamiento de datos. Las entidades son representadas en forma de documentos orientados y es no relacional [43].

La solución propuesta cuenta con 11 documentos, Se tuvo en cuenta el patrón de diseño de base de datos, llave subrogada, el cual propone la generación de una llave primaria única para cada entidad en vez de usar un atributo identificador en el contexto dado. MongoDB asigna el id automáticamente del tipo ObjectId, una combinación de números y letras [44]. A continuación se muestra un fragmento de la estructura de los documentos.

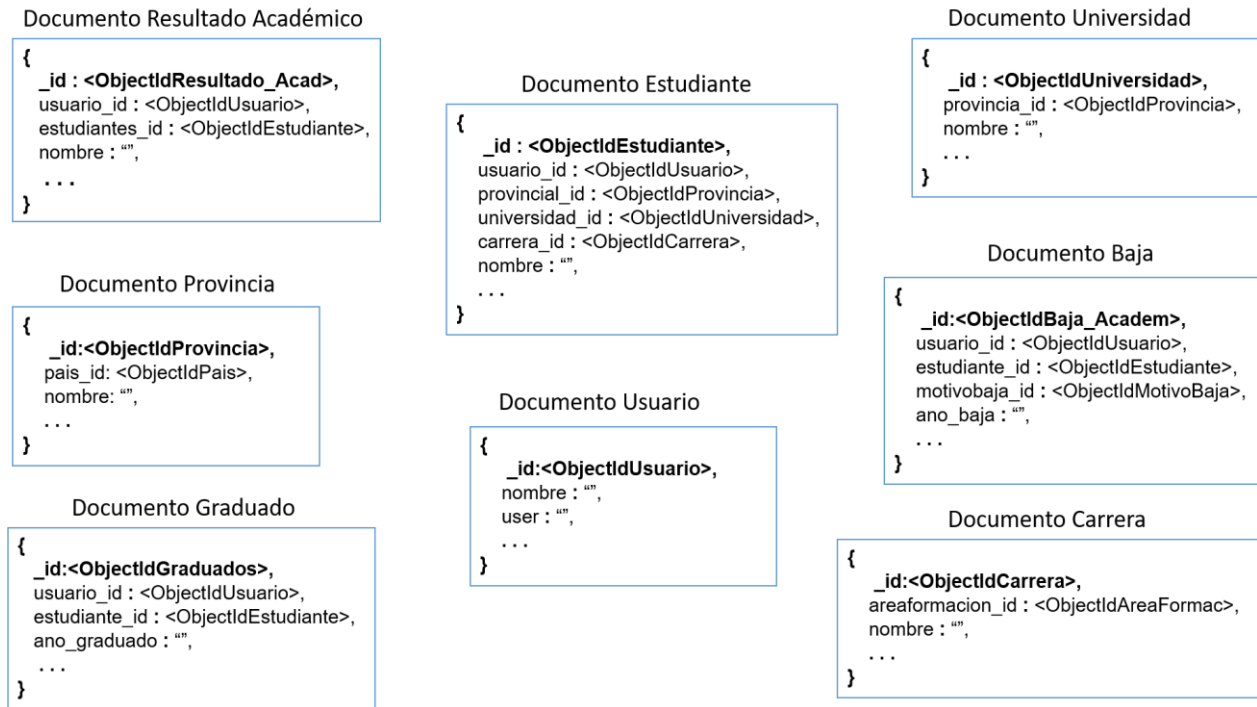


Figura 9. Modelo de datos.

MongoDB siendo una base de datos NoSQL donde los documentos no tienen relación entre ellos, pero los contenidos de los documentos se pueden relacionar con los contenidos de otros documentos por el id. [44]. Es evidente la relación en el documento estudiante en algunas entidades que reciben los id de contenido de otros documentos. Abajo una imagen que ejemplifica como es una relación en MongoDB.

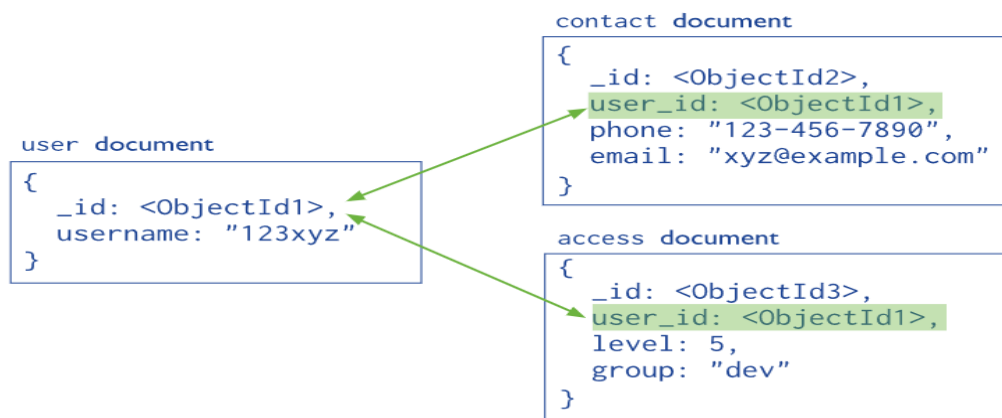


Figura 10. Ejemplo de relación en MongoDB [45].

3.1.2 Tareas de Ingeniería

En la fase de Planeación se definieron los elementos funcionales necesarios para realizar el desarrollo del sistema, lo que implica que ya está disponible para continuar a la fase de implementación, donde Tareas de Ingeniería (TI) es el artefacto usado [46]. Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

A continuación, se muestran algunas de las tareas de ingeniería correspondientes a cada iteración, las demás tareas correspondientes pueden ser consultadas en el anexo 3.

Primera iteración se definieron las siguientes tareas de ingeniería:

HU 1: Adicionar Estudiante

- TI 1: Crear el servidor de servicios API
- TI 2: Crear la base de la estructura de la interfaz de usuario
- TI 3: Adicionar Estudiante

HU 2: Listar Estudiantes

- TI 4: Listar Estudiantes

HU 3: Ver Estudiante

- TI 5: Ver Estudiante

HU 4: Dar de Baja a un Estudiante

- TI 6: Dar de Baja a un Estudiante

HU 5: Buscar Por Nombre

- TI 7: Buscar Por Nombre

HU 6: Graduar Estudiante

- TI 8: Graduar Estudiante

HU 7: Transitar de Clase a un Estudiante

- TI 9: Transitar de Clase a un Estudiante

HU 8: Editar Estudiante

- TI 10: Editar Estudiante

HU 9: Listar Estudiantes dados de Baja

- TI 11: Listar Estudiantes dados de Baja

HU 10: Listar Estudiantes Graduados

- TI 12: Listar Estudiantes Graduados

HU 11: Reintegrar Estudiante dado de Baja

- TI 13: Reintegrar Estudiante dado de Baja

HU 12: Gestionar Resultado Académico

- TI 14: Gestionar Resultado Académico

Tabla 38. Tarea de ingeniería crear el servidor de servicio API REST

Tarea de ingeniería	
No. tarea: 1	Historia de Usuario 1: Adicionar Estudiante
Nombre tarea: Crear el servidor de servicios API REST	
Tipo de tarea: Desarrollo	Punto estimado: 1
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar el API REST en Node JS y Express JS necesarias para desarrollar el software. Crear los modelos y las rotas necesarias para el funcionamiento inicial del software	

Tabla 40. Tarea de ingeniería adicionar estudiante

Tarea de ingeniería	
No. tarea: 3	Historia de Usuario 1: Adicionar Estudiante
Nombre tarea: Adicionar Estudiante	
Tipo de tarea: Desarrollo	Punto estimado: 1

Programador responsable: Pedro M. Nekaka

Descripción: Desarrollar la interfaz de usuario para permitir adicionar un estudiante

Segunda iteración se definieron las siguientes tareas de ingeniería:

HU 13: Gestionar Provincia

- TI 15: Gestionar Provincia

HU 14: Gestionar Universidad

- TI 16: Gestionar Universidad

HU 15: Gestionar Área de Formación

- TI 17: Gestionar Área de Formación

HU 16: Gestionar Carrera

- TI 18: Gestionar Carrera

Tabla 52. Tarea de ingeniería gestionar universidad

Tarea de ingeniería	
No. tarea: 16	Historia de Usuario 14: Gestionar Universidad
Nombre tarea: Gestionar Universidad	
Tipo de tarea: Desarrollo	Punto estimado: 2
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interfaz de usuario para permitir adicionar, editar, listar y eliminar universidad en que los estudiantes cursan las carreras.	

Tercera iteración se definieron las siguientes tareas de ingeniería:

HU 17: Gestionar País

- TI 19: Gestionar País

HU 18: Buscar por Filtros

- TI 20: Buscar por Filtros

HU 19: Gestionar Motivo de Baja

- TI 21: Gestionar Motivo de Baja

Tabla 56. Tarea de ingeniería buscar por filtros

Tarea de ingeniería	
No. tarea: 20	Historia de Usuario 18: Buscar por Filtros
Nombre tarea: Buscar por Filtros	
Tipo de tarea: Desarrollo	Punto estimado: 5
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar métodos para permitir que las interfaces donde se requiere un filtro funcionen según lo deseado. Los métodos de filtro son desarrollados en el servidor de API REST y otros en la interfaz de usuario en React JS.	

Cuarta iteración se definieron las siguientes tareas de ingeniería:

HU 20: Gestionar Usuario

- TI 22: Gestionar Usuario
- TI 23: Login
- TI 24: Perfil del Usuario

Tabla 58. Tarea de ingeniería gestionar usuario

Tarea de ingeniería	
No. tarea: 22	Historia de Usuario 20: Gestionar Usuario
Nombre tarea: Gestionar Usuario	
Tipo de tarea: Desarrollo	Punto estimado: 3
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interfaz de usuario y los métodos para permitir adicionar, editar, listar y eliminar los usuarios del sistema.	

3.2 Pruebas

Debido al aumento del tamaño y la complejidad del software, el proceso de prueba se ha convertido en una tarea vital dentro del proceso de desarrollo de cualquier sistema. Los ingenieros de software garantizan la calidad aplicando métodos técnicos sólidos y medidos, llevando a cabo pruebas de software bien planificadas [47].

El proceso de pruebas de software tiene dos objetivos fundamentales:

1. Demostrar al desarrollador y al cliente que el software satisface sus requisitos.
2. Descubrir defectos en el software en que el comportamiento de este es incorrecto, no deseable o no cumple su especificación [48].

La metodología de desarrollo empleada para el trabajo de diploma divide las pruebas del sistema en dos grupos: pruebas unitarias, y pruebas de aceptación o pruebas.

3.2.1 Pruebas Unitarias

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código [49].

Node JS utiliza las librerías Mocha y Chai para realizar las pruebas, la cual es uno de los entornos para realizar pruebas unitarias en el lenguaje de programación JavaScript. Se creó con la idea de que cuanto antes se detecten los errores en el código antes podrán ser corregidos. Es de fácil uso y se ejecuta automáticamente una vez bien configurado en la aplicación que se estará desarrollando.

Hay que crear una carpeta con el nombre test, donde se pone los ficheros creados. A continuación, se presenta parte del código de prueba de la API REST usuario:

```

22 // GET - Listar todos los usuarios
23 it('debe retornar todos los usuario', function() {
24     return chai.request(app)
25         .get(basePath + '/usuario')
26         .then(function(res) {
27             expect(res).to.have.status(200);
28             expect(res).to.be.json;
29             expect(res.body).to.be.an('array');
30         });
31 });
32
33 // GET - Listar un usuario
34 it('debe retornar un usuario', function() {
35     return chai.request(app)

```

Figura 11. Prueba de la API REST usuario.

Para Probar lo deseado basta con escribir en la terminal ubicándose en la carpeta raíz del proyecto y ejecutar el comando `npm test`, Mocha va a localizar todos los ficheros de teste encontrados en la carpeta `test`. El resultado se muestra una notificación de éxito en la consola si el fichero de teste este bien implementado y la API REST, el método o la fundición este bien implementado, caso contrario se presenta una notificación de fallo. A continuación, se presenta la evidencia de lo mencionado anteriormente:

```

Inicio de Teste API Usuario
GET /api/usuario 404 7.324 ms - 80
  1) debe retornar todos los usuario
GET /api/usuario/1 404 0.750 ms - 82
  2) debe retornar un usuario
POST /api/usuario 404 15.424 ms - 80
  3) para poder adicionar nuevo usuario

```

Figura 12. Prueba de la API REST usuario, notificación de fallo.


```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Inicio de Teste API Usuario
GET /api/usuario 200 6.741 ms - 163
  ✓ debe retornar todos los usuario (39ms)
GET /api/usuario/1 200 1.128 ms - 81
  ✓ debe retornar un usuario
POST /api/usuario 201 19.023 ms - 8
  ✓ para poder adicionar nuevo usuario
PS E:\node-unit-test-master>

```

Figura 13. Prueba de la API REST usuario, notificación de éxito.

3.2.2 Pruebas de Aceptación

Las pruebas de aceptación son creadas a partir de las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada.

Las pruebas de aceptación son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información [48].

A continuación, se presentan 5 de los 20 casos de pruebas de aceptación, los cuales fueron realizados a las historias de usuario, el resto de los casos de pruebas de aceptación se pueden encontrar en el anexo 4:

Tabla 61. Caso de Prueba de aceptación HU1_P1 para la HU “Adicionar estudiante”

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario 1: Adicionar Estudiante
Nombre: Adicionar Estudiante	
Descripción: Prueba la funcionalidad de registrar un estudiante.	
Condiciones de ejecución:	

El usuario debe estar autenticado.

El usuario debe estar en el módulo estudiante.

Para adicionar un estudiante el usuario debe verificar si la provincia, carrera, universidad del estudiante ya se encuentra adicionado en el sistema.

Pasos de ejecución:

El usuario debe seleccionar en el botón “Novo Estudiante” se muestra la interfaz con un formulario, que será usado para añadir un nuevo estudiante. Para tal, el usuario debe asegurarse que los datos entrados al sistema sean correctos y evitar dejar campos vacíos.

El usuario debe presionar en “Guardar”

Resultados esperados:

El sistema muestra un mensaje de éxito.

El sistema registra un nuevo evento si los datos fueron proporcionados correctamente. En otro caso muestra el mensaje de error correspondiente.

Evaluación de la prueba: Prueba satisfactoria.

Tabla 62. Caso de Prueba de aceptación HU2_P2 para la HU “Listar estudiante”

Caso de Prueba de Aceptación	
Código: HU2_P2	Historia de Usuario 2: Listar Estudiantes
Nombre: Listar Estudiante	
Descripción: Prueba la funcionalidad de Listar estudiantes.	
Condiciones de ejecución:	
El usuario debe estar autenticado.	
Es necesario tener estudiantes registrados en el sistema.	
Pasos de ejecución:	

El usuario debe ir al módulo estudiantes y escoger entre estudiante de postgrado o licenciatura, luego se listan los datos de los estudiantes en una tabla

Resultados esperados:

El sistema busca en la base de datos los estudiantes, si existe se muestra en una tabla caso contrario, muestra una tabla sin registro.

Evaluación de la prueba: Prueba satisfactoria.

Tabla 63. Caso de Prueba de aceptación HU3_P3 para la HU “Ver estudiante”

Caso de Prueba de Aceptación	
Código: HU3_P3	Historia de Usuario 3: Ver Estudiantes
Nombre: Ver Estudiante	
Descripción: Prueba la funcionalidad de ver un estudiante.	
Condiciones de ejecución: El usuario debe estar autenticado. Es necesario tener el estudiante registrado en el sistema.	
Pasos de ejecución: El usuario debe ir al módulo estudiantes y escoger entre estudiante de postgrado o licenciatura, después en el listado de estudiantes debe seleccionar el estudiante y selecciona en visualizar, botón que se torna visible después que se haga la selección del estudiante.	
Resultados esperados: El sistema muestra una interfaz con toda información del estudiante seleccionado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 64. Caso de Prueba de aceptación HU4_P4 para la HU “Dar de baja”

Caso de Prueba de Aceptación	
Código: HU4_P4	Historia de Usuario 4: Dar de Baja
Nombre: Dar de Baja	
Descripción: Prueba la funcionalidad de dar de baja a un estudiante.	
Condiciones de ejecución:	
El usuario debe estar autenticado.	
El usuario debe ser de nivel 0 para poder dar de baja a un estudiante.	
Es necesario tener el estudiante registrado en el sistema.	
Pasos de ejecución:	
El usuario debe ir al módulo estudiantes y escoger entre estudiante de postgrado o licenciatura, después en el listado de estudiantes debe seleccionar el estudiante y presionar en visualizar, botón que se torna visible después que se haga la selección del estudiante, después puede presionar en el botón "Baja".	
El usuario debe llenar el formulario que se le mostrara y presionar en ok.	
Resultados esperados:	
El sistema muestra una interfaz de confirmación, si el usuario confirma, se le muestra una notificación de éxito, si no, se cancela la operación.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 65. Caso de Prueba de aceptación HU5_P5 para la HU "Búsqueda por nombre"

Caso de Prueba de Aceptación	
Código: HU5_P5	Historia de Usuario 5: Búsqueda por nombre
Nombre: Búsqueda por nombre	
Descripción: Prueba la funcionalidad de búsqueda por nombre.	

Condiciones de ejecución:

El usuario debe estar autenticado.

Es necesario tener el estudiante registrado en el sistema.

Al realizar la búsqueda el usuario debe escribir el nombre lo máximo posible, pues el resultado se mostrará mientras se escribe letra por letra

Pasos de ejecución:

El usuario debe ir al módulo estudiantes y escoger entre estudiante de postgrado, graduados, verificar baja o licenciatura, luego se listan los datos de los estudiantes en una tabla, luego en una caja de texto debe escribir el nombre deseado.

Resultados esperados:

El sistema muestra el resultado de la búsqueda en la tabla caso encuentra correspondencia a lo que se escribe en la caja de texto.

Evaluación de la prueba: Prueba satisfactoria.

3.2.3 Pruebas de Seguridad

Según Pressman las pruebas de seguridad intentan verificar que los mecanismos de protección incorporados en el sistema lo protegerán de accesos inapropiados. Durante las pruebas de seguridad, el responsable de la prueba desempeña el papel de un individuo que desea entrar en el sistema. Debe intentar conseguir las claves de acceso por cualquier medio, debe bloquear el sistema, negando así el servicio a otras personas [47].

Resultados de las pruebas de seguridad

Con el objetivo de evaluar la seguridad de la solución propuesta se emplea la herramienta OWASP ZAP 2.7.0 la cual arrojó los siguientes resultados luego de realizada una primera iteración.

Tabla 81. Resultados de las pruebas de seguridad.

Categorías de vulnerabilidades	Cantidad de errores
Credenciales de usuarios enviadas en texto plano	4
Vínculos rotos	1

Campos de contraseña con auto completamiento activado	2
Campos de usuario y contraseña mostrados	0
Total	7

Después de analizar los resultados obtenidos en las pruebas se procedió a corregir las deficiencias encontradas. Para ello se llevaron a cabo un conjunto de acciones que permitió reforzar la seguridad del sistema.

Campos de contraseña con auto completamiento activado, cuando se introduce un nuevo nombre y contraseña en un formulario y se envía el formulario, el navegador le pregunta si la contraseña debe ser guardada. Cuando se muestra el formulario, el nombre y la contraseña se rellenan de forma automática o se completan como se introduce el nombre. Un atacante con acceso local podría obtener la contraseña de texto plano de la caché del navegador. Para darle solución, la función de la contraseña de autocompletar debe ser desactivada. Para desactivar la función de autocompletar, es posible utilizar un código similar al siguiente: `<INPUT TYPE="password" AUTOCOMPLETE="off">`.

3.3 Implementación y validación del sistema

Concluido el desarrollo del software, el mismo fue implantado en el Sector de Apoyo Estudiantil en la embajada de Angola en Cuba en el día 02 de febrero de 2020. El ambiente de la aplicación cumple con estándares de diseño, con el fin de hacerlo familiar e intuitivo para los usuarios que interactúan con el mismo ya que los usuarios finales no son de la rama de la informática.

Para la validación de la aplicación desarrollada se toma en cuenta la variable gestión de la información de los procesos asociados a los estudiantes.

Antes de la implementación del sistema propuesto, SAE utilizaba un sistema hecho en Access que permitía registrar estudiantes, editar, buscar por nombre y eliminar estudiantes, además permitía registrar el resultado académico del año que culmina. El sistema antiguo tenía muchas limitaciones para la gestión de la información de los procesos asociados a los estudiantes en el SAE, falta de registro de estudiantes dados de baja, falta de registro de estudiantes graduados en años anteriores, separación de estudiantes de postgrado y de graduación, la integridad y confidencialidad de la información no se aplicaba.

Actualmente con el sistema propuesto implementado, contribuye en la gestión de la información de los procesos asociados a los estudiantes:

- Con el sistema de autenticación se garantiza que solo el personal autorizado tenga acceso a la información e solo el personal autorizado puede modificar la información de los estudiantes. Además, con la asignación de roles o nivel de acceso permite que los usuarios puedan hacer lo que le es permitido.
- Registrar estudiantes, Editar, listar estudiantes, dar de baja académica o graduar una vez terminada su carrera. Registrar los resultados académicos de los estudiantes de los años anteriores, permitiendo hacer una trayectoria académica de los estudiantes.
- El SAE ahora cuenta con el registro de los estudiantes graduados de los años anteriores, precisamente los graduados en el curso 2018-2019 y registro de los estudiantes dado de baja en 2019 a 2020.
- Buscar los estudiantes en el sistema ahora es fácil y con mayor precisión. El sistema permite almacenar cantidades grandes de información según la capacidad de almacenamiento de la computadora.

3.4 Conclusiones del capítulo

En el capítulo se expuso los principales resultados obtenidos una vez realizadas las tareas 5 y 6 de la investigación, a partir de lo cual se concluye que:

- El modelo de datos proporcionó una guía para la generación de las entidades y para la especificación de sus relaciones, facilitando la creación de un esquema de base de datos de 11 documentos.
- La implementación de las 24 tareas de ingeniería permitió que fuera posible el cumplimiento del plan de entrega de las historias de usuario.
- La realización de pruebas de aceptación, unitarias y de seguridad permitió comprobar que las funcionalidades fueron implementadas correctamente y que el sistema satisface las necesidades del cliente. La prueba de seguridad garantiza que el sistema cumple con el estándar base de seguridad.

CONCLUSIONES GENERALES

A través del desarrollo del presente trabajo de diploma se les dio cumplimiento a las tareas de investigación definidas al inicio del mismo, teniendo como principal resultado el Sistema de Gestión de Estudiantes para el Sector de Apoyo Estudiantil de Angola en Cuba. Luego del análisis de los resultados obtenidos se concluye que:

El estudio de los sistemas similares, arrojó que ninguno brindaba una solución completa a las particularidades del Sector de Apoyo Estudiantil de Angola en Cuba, lo cual constituyó motivo para el desarrollo de la investigación realizada. Se decidió desarrollar la implementación de la solución con las herramientas y tecnologías descritas en el capítulo 1: React JS, Node JS, Mongo DB, Express, Visual Studio Code, entre otras, siguiendo como metodología de desarrollo de software XP.

La especificación de las historias de usuario y de los requisitos no funcionales del sistema, posibilitó una descripción detallada del software que sirvió de base para diseñar la propuesta de solución. El empleo del patrón arquitectónico MVC posibilitó tener separados los datos, la interfaz de usuario y la lógica de negocio en tres componentes distintos, facilitando la implementación y mantenimiento del sistema.

La aplicación de los estándares de codificación permitió una implementación responsable y organizada facilitando la obtención del sistema propuesto, además de garantizar la posibilidad de mantenimientos futuros del mismo. Para verificar el correcto funcionamiento de la solución obtenida se aplicaron pruebas de aceptación, pruebas unitarias y de seguridad, además con las pruebas se pudo verificar la satisfacción de las necesidades del cliente.

Se obtuvo un sistema web funcional, que contribuye en la gestión de la información de estudiantes en el Sector de Apoyo Estudiantil cumpliendo así el objetivo de la investigación, además se evidencia la validación de la solución propuesta en el capítulo 3. El sistema implementado ayuda al INAGBE a planificar mejor el regreso de los estudiantes graduados. Además, se comprobó que en el nuevo sistema no existirá más pérdida de información por problema de almacenamiento.

RECOMENDACIONES

- Se recomienda el desarrollo de un módulo para la gestión de procesos que tiene que ver con los estipendios que los estudiantes reciben, con los pagos de las carreras y los seguros de salud.
- También es importante hacer unos cambios en configuraciones de los usuarios del sistema, para que el sistema sea implementado en Angola a fin de que todos los SAE en el mundo puedan tener un único sistema y que solo puedan tener acceso a la información de los estudiantes que se encuentran en el país correspondiente.
- Se recomienda también la creación de un módulo para gestionar los resultados académicos, donde las universidades puedan tener acceso al módulo para gestionar los resultados académicos.

REFERENCIA BIBLIOGRÁFICA

- [1] ISO 27001 - Software ISO 27001 de Sistemas de Gestión. [en línea], [sin fecha]. [Consulta: 19 junio 2020]. Disponible en: <https://www.isotools.org/normas/riesgos-y-seguridad/iso-27001/>.
- [2]- Ministério do Ensino Superior, Ciência, Tecnologia e Inovação | Detalhes. [en línea], [sin fecha]. [Consulta: 10 junio 2020]. Disponible en: <https://www.mescti.gov.ao/VerLegislacao.aspx?id=2404>.
- [3]- ASALE, R.- y RAE, [sin fecha]. información | Diccionario de la lengua española. «Diccionario de la lengua española» - Edición del Tricentenario [en línea]. [Consulta: 18 junio 2020]. Disponible en: <https://dle.rae.es/información>.
- [4]- WARFIELD, J.N., 1975. A Review of: "THEORETICAL ANALYSIS OF INFORMATION SYSTEMS", Fourth Edition, by Börje Langefors. Studentlitteratur, Sweden and Auerbach Publishers, Philadelphia, 1973. 489 pp. International Journal of General Systems, vol. 2, no. 1, pp. 122-123. ISSN 0308-1079. DOI 10.1080/03081077508960850.
- [5]- Capítulo I Gestión De La Información Y Conocimiento: Por Phil Bartle, Phd Traducción De Ma Lourdes Sada. Scribd [en línea], [sin fecha]. [Consulta: 18 junio 2020]. Disponible en: <https://www.scribd.com/document/55030606/Gestion-de-ion-Conocimiento>.
- [6]- FERRERA, P. y YENISLEIDY, A., 2017. Módulo Ventas del Sistema de Gestión para la Agencia de Renta de Vehículos REX. [en línea], [Consulta: 22 febrero 2020]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/8032>.
- [7]- Sistema informático - EcuRed. [en línea], [sin fecha]. [Consulta: 10 junio 2020]. Disponible en: https://www.ecured.cu/Sistema_inform%C3%A1tico.
- [8]- USO, B. de D. de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior, [sin fecha]. Bases de datos - EcuRed. [en línea]. [Consulta: 10 junio 2020]. Disponible en: https://www.ecured.cu/Bases_de_datos.
- [9]- BRÁS, C.A. y DIVOVO, M.D., 2018. Ensino Superior em Angola: realidade e desafios na 4ª República. UDZIWI: Revista da Educação [en línea], vol. 0, no. 28. [Consulta: 2 diciembre 2019]. Disponible en: <https://www.revista.up.ac.mz/index.php/UDZIWI/article/view/287>.

- [10]- Listagem das Instituições de Ensino Superior privadas reconhecidas. [en línea], [sin fecha]. [Consulta: 2 diciembre 2019]. Disponible en: <https://barbante.blogs.sapo.ao/listagem-das-instituicoes-de-ensino-17257>.
- [11]- RODRÍGUEZ PALMERO, Y., SAAVEDRA REMÓN, L.E., CIUDAD RICARDO, F.A. y SERRANO PÉREZ, W., 2016. Subsistema para la gestión de la información del sistema automatizado para la Superación Pedagógica del Claustro de profesores en la Universidad de las Ciencias Informáticas. [en línea], [Consulta: 5 diciembre 2019]. Disponible en: <http://repositorio.uci.cu/jspui/handle/ident/9141>.
- [12]- Akademos, un Sistema Automatizado para la Gestión Académica | Serie Científica de la Universidad de las Ciencias Informáticas. [en línea], [sin fecha]. [Consulta: 19 junio 2020]. Disponible en: <https://publicaciones.uci.cu/index.php/serie/article/view/253>.
- [13]- Ingeniería del Software Ingeniería del Software de Gestión. Tema 3 Metodologías de Desarrollo de Software - PDF Free Download. [en línea], [sin fecha]. [Consulta: 21 febrero 2020]. Disponible en: <https://docplayer.es/31788698-Ingenieria-del-software-ingenieria-del-software-de-gestion-tema-3-metodologias-de-desarrollo-de-software.html>.
- [14]- (PDF) METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES. ResearchGate [en línea], [sin fecha]. [Consulta: 21 febrero 2020]. Disponible en: https://www.researchgate.net/publication/299506242_METODOLOGIAS_TRADICIONALES_VS_METODOLOGIAS_AGILES.
- [15]- MONTERO, B.M., CEVALLOS, H.V. y CUESTA, J.D., 2018. Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. Espirales Revista Multidisciplinaria de investigación [en línea], vol. 2, no. 17. [Consulta: 3 marzo 2020]. ISSN 2550-6862. DOI 10.31876/re.v2i17.269. Disponible en: <http://revistaespirales.com/index.php/es/article/view/269>.
- [16]- Roger S. Pressman, Ingeniería del Software un enfoque práctico, Séptima edición. [Capítulo 3: Desarrollo ágil].
- [17]- JENNIFER V. LÓPEZ ÁLAVA, 2014. Ingeniería del Software: Tema 1. Metodologías de Desarrollo de Software: Metodologías Ágiles. Ingeniería del Software [en línea]. [Consulta: 3 marzo 2020]. Disponible en: <http://ingsoftwarejlo.blogspot.com/2014/11/normal-0-21-false-false-false-es-ec-x.html>.

- [18]- BHANDARI, D., 2019. Job Vacancy Application using MERN stack. [en línea]. [Consulta: 4 diciembre 2019]. Disponible en: <http://www.theseus.fi/handle/10024/170363>.
- [19]- La base de datos líder del mercado para aplicaciones modernas. MongoDB [en línea], [sin fecha]. [Consulta: 10 junio 2020]. Disponible en: <https://www.mongodb.com/es>.
- [20]- CASTILLO, J.N., GARCÉS, J.R., NAVAS, M.P., SEGOVIA, D.F.J. y NARANJO, J.E.A., 2017. Base de Datos NoSQL: MongoDB vs. Cassandra en operaciones CRUD (Create, Read, Update, Delete). Revista Publicando, vol. 4, no. 11(1), pp. 79-107. ISSN 1390-9304.
- [21]- React – Una biblioteca de JavaScript para construir interfaces de usuario. [en línea], [sin fecha]. [Consulta: 10 junio 2020]. Disponible en: <https://es.reactjs.org/>.
- [22]- NODE.JS, [sin fecha]. Node.js. Node.js [en línea]. [Consulta: 10 junio 2020]. Disponible en: <https://nodejs.org/es/>.
- [23]- NEVALAINEN, T., 2018. MEAN Software Stack: MEAN-stack and its relative strengths and weaknesses relative to its competitors. [en línea]. [Consulta: 4 diciembre 2019]. Disponible en: <http://www.theseus.fi/handle/10024/154731>.
- [24]- LINDÉN, L., 2019. AngularJS-sovelluksen päivittäminen Angular-sovellukseksi. [en línea]. [Consulta: 4 diciembre 2019]. Disponible en: <http://www.theseus.fi/handle/10024/260986>.
- [25]- HAUTAVIITA, A., 2018. Developing a web application on the MEVN stack: the architecture of a full-stack JavaScript application. [en línea]. [Consulta: 4 diciembre 2019]. Disponible en: <http://www.theseus.fi/handle/10024/156235>.
- [26]- ROBLEDO CABRERIZO, F., 2019. Desarrollo de aplicación SPA en REACT, apoyada en API RESTFull para la escucha de emisoras de radio online. [en línea], [Consulta: 4 diciembre 2019]. Disponible en: <https://ebuah.uah.es/dspace/handle/10017/39530>.
- [27]- PETUKHOVA, E., 2019. Sitecore JavaScript Services Framework Comparison.
- [28]- GAUCHAT, J.D., 2012. El gran libro de HTML5, CSS3 y JavaScript. S.l.: Marcombo. ISBN 978-84-267-1782-5.
- [29]- La base de datos líder del mercado para aplicaciones modernas. MongoDB [en línea], [sin fecha]. [Consulta: 10 junio 2020]. Disponible en: <https://www.mongodb.com/es>.

- [30]- Documentation for Visual Studio Code. [en línea], [sin fecha]. [Consulta: 10 junio 2020]. Disponible en: <https://code.visualstudio.com/docs>.
- [31]- OWASP ZAP Zed Attack Proxy | OWASP. [en línea], [sin fecha]. [Consulta: 10 junio 2020]. Disponible en: <https://owasp.org/www-project-zap/>.
- [32]- ANDRY, J.F., 2019. - Development of Heavy Equipment Rental System Using eXtreme Programming Method. IJNMT (International Journal of New Media Technology), vol. 6, no. 1, pp. 24-31. ISSN 2581-1851. DOI 10.31937/ijnmt.v6i1.966.
- [33]- Faiza Anwer, Shabib Aftab, y Iftikhar Ali, 2017. Proposal of Tailored Extreme Programming Model for Small Projects. [en línea]. [Consulta: 23 febrero 2020]. Disponible en: https://www.researchgate.net/profile/Shabib_Aftab/publication/319162622_Proposal_of_Tailored_Extreme_Programming_Model_for_Small_Projects/links/5a082783a6fdcc65eab3d0e4/Proposal-of-Tailored-Extreme-Programming-Model-for-Small-Projects.pdf.
- [34]- Faiza Anwer, Shabib Aftab, Syed Shah Muhammad Shah and Usman Waheed. Comparative Analysis of Two Popular Agile Process Models: Extreme Programming and Scrum. Department of Computer Science and Virtual University of Pakistan [Consulta: 23 febrero 2020]. Disponible en: https://www.researchgate.net/profile/Shabib_Aftab/publication/316845761_Comparative_Analysis_of_Two_Popular_Agile_Process_Models_Extreme_Programming_and_Scrum/links/5913588fa6fdcc963e7ee052/Comparative-Analysis-of-Two-Popular-Agile-Process-Models-Extreme-Programming-and-Scrum.pdf.
- [35]- LETELIER, P. y PENADÉS, M.C., 2006. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). www.cyta.com.ar/ta0502/v5n2a1.htm [en línea]. [Consulta: 23 febrero 2020]. Disponible en: <https://www.cyta.com.ar/ta0502/v5n2a1.htm>.
- [36]- Ian Sommerville. Ingeniería del Software. 2005. Séptima edición. [Capítulo 6: Requisitos del Software].
- [37]- Larman C., UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado, Segunda edición, 2003 [en línea]. [Consulta: 8 mayo 2020]. Disponible en: <http://fmonje.com/UTN/ADES%20-%202008/UML%20y%20Patrones%20%202da%20Edicion.pdf>.

- [38]- Giraldo G., Gloria L.; AcevedoO., Juan F.; Moreno N., David A. Una ontología para la representación de diseño de software. [en línea]. [Consulta: 8 mayo 2020]. Disponible en: <https://www.redalyc.org/pdf/1331/133122679013.pdf>.
- [39]- GUERRERO, Carlos A; SUAREZ, Johanna M y GUTIERREZ, Luz E. Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. Inf. tecnol. [online]. 2013, vol.24, n.3 [citado 2020-05-08], pp.103-114. Disponible en: <https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-07642013000300012&lng=es&nrm=iso>. ISSN 0718-0764. <http://dx.doi.org/10.4067/S0718-07642013000300012>.
- [40]- Francisco José. Patrones de Alexander a la tecnología de objetos. [en línea]. [Consulta: 8 mayo 2020]. Disponible en: <https://repositorio.grial.eu/bitstream/grial/356/1/patrones1.pdf>.
- [41]- Abraham SilbSillbererschatz y Henry F. Korth, Database System Concepts, Sexta. McGraw-Hill, 2015.
- [42]- Evan M.Hahn. Express in Action – writing, building and testing node.js applications. [en línea]. [Consulta: 27 marzo 2020]. Disponible: <http://www.hackerstribes.com/wp-content/uploads/2016/04/Node.js-Express-in-Action.pdf>.
- [43]- Li Liang, Ligu Zhu, Wenqian Shang, Dongyu Feng, Zida Xiao. Express Supervision System Base on NodeJs and MongoDB. [en línea]. [Consulta: 27 marzo 2020]. Disponible en: <http://ieeexplore.ieee.org/abstract/document/7960064/>.
- [44]-Data Modeling Introduction. [en línea]. [Consulta: 11 mayo 2020].Disponible en: <https://docs.mongodb.com/manual/core/data-modeling-introduction/>.
- [45]-Data Model Design. [en línea]. [Consulta: 11 mayo 2020].Disponible en: <https://docs.mongodb.com/manual/core/data-model-design/>.
- [46]- Andy Hernández Paez. Ingeniería de Software aplicada a los trabajos de diploma. Facultad 4. Universidad de las Ciencias Informáticas.
- [47]- Roger S. Pressman. Ingeniería del Software un enfoque práctico. Séptima edición. [Capítulo 17: Estrategias de Prueba de software].
- [48]- Ian Sommerville. Ingeniería del Software. 2005. Séptima edición. [Capítulo 23: Pruebas del Software].

[49]- José Joskowicz. Reglas y Practicas en eXtreme Programming. [en linea]. [Consulta: 16 mayo 2020].
Disponible en: https://www.academia.edu/download/31398587/xp_-_jose_joskowicz.pdf.

ANEXO

Anexo 1: Historias de usuarios.

Tabla 8. HU-4 Listar Estudiantes

Historia de Usuario 2	
Nombre: Listar Estudiantes	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alta
Puntos Estimados: 1	Puntos Reales: 2
Descripción: El usuario debe ir al módulo estudiantes y escoger entre estudiante de postgrado o licenciatura, luego se listan los datos de los estudiantes en una tabla: La imagen del estudiante, el nombre, sexo, año académico, universidad, provincia y carrera.	
Observación:	

Tabla 6. HU-2 Ver Estudiante

Historia de Usuario 3	
Nombre: Ver Estudiante	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Media
Puntos Estimados: 1	Puntos Reales: 1
Descripción: El usuario debe ir al módulo estudiantes y escoger entre estudiante de postgrado o licenciatura, después en el listado de estudiantes debe seleccionar el estudiante y presionar en visualizar. botón que se torna visible después que se haga la selección de uno o más estudiantes	
Observación: Todos los usuarios tienen privilegio para ver estudiante. Es necesario filtrar el estudiante deseado en el caso de no ser visible de antemano.	

Tabla 9. HU-5 Búsqueda por nombre

Historia de Usuario 5

Nombre: Búsqueda por nombre	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Media
Puntos Estimados: 1	Puntos Reales: 1
Descripción: El usuario debe ir al módulo estudiantes y escoger entre estudiante de postgrado, graduados, verificar baja o licenciatura, luego se listan los datos de los estudiantes en una tabla, luego en una caja de texto debe escribir el nombre deseado.	
Observación: Al realizar la búsqueda el usuario debe escribir el nombre lo máximo posible, pues el resultado se mostrará mientras se escribe letra por letra.	

Tabla 10. HU-6 Graduar Estudiante

Historia de Usuario 6	
Nombre: Graduar Estudiante	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Puntos Estimados: 1	Puntos Reales: 1
Descripción: El usuario debe ir al módulo estudiantes y escoger entre estudiante de postgrado o de licenciatura, luego se listan los datos de los estudiantes en una tabla, se selecciona el estudiante o los estudiantes en seguida presionar en graduar, botón que se torna visible después que se haga la selección de uno o más estudiantes	
Observación: En el apartado de visualizar estudiante también la funcionalidad está disponible	

Tabla 11. HU-7 Transitar de Clase a un Estudiante

Historia de Usuario 7	
Nombre: Transitar de Clase a un Estudiante	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Puntos Estimados: 1	Puntos Reales: 2

Descripción: El usuario debe ir al módulo estudiantes y escoger entre estudiante de postgrado o de licenciatura, luego se listan los datos de los estudiantes en una tabla, se selecciona el estudiante o los estudiantes en seguida presionar en transitar, botón que se torna visible después que se haga la selección de uno o más estudiantes. Solo transita de clase, o año académico los estudiantes que no son finalistas. Al transitar estudiantes, automáticamente se debe también adicionar los resultados académicos, donde los campos media académica y cantidad de arrastre deben ser (-), pero el año académico = al año que cursa, resultado = aprobado, curso = al curso lectivo que culmino.

Observación: La funcionalidad solo está disponible en la tabla de listar los estudiantes.

Tabla 12. HU-8 Editar Estudiante

Historia de Usuario 8	
Nombre: Editar Estudiante	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Media
Puntos Estimados: 2	Puntos Reales: 3
Descripción: El usuario debe ir al módulo estudiantes y escoger entre estudiante de postgrado, graduados, verificar baja o de licenciatura, luego se listan los datos de los estudiantes en una tabla, se selecciona el estudiante en seguida presionar en editar, botón que se torna visible solo después que se haga la selección de uno estudiantes.	
Observación: La funcionalidad está disponible también en el apartado visualizar estudiantes.	

Tabla 13. HU-9 Listar Estudiante dado de Baja

Historia de Usuario 9	
Nombre: Listar estudiante dado de Baja	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Media
Puntos Estimados: 1	Puntos Reales: 1

Descripción: El usuario debe ir al módulo estudiantes y escoger verificar baja, luego se listan los datos de los estudiantes en una tabla, si selecciona un estudiante en seguida tendrá el botón visualizar.

Observación: Al presionar en el botón visualizar tendrá opciones como editar estudiante o adicionar resultado académico disponible.

Tabla 14. HU-10 Listar Estudiante Graduados

Historia de Usuario 10	
Nombre: Listar estudiante Graduados	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Media
Puntos Estimados: 1	Puntos Reales: 1
Descripción: El usuario debe ir al módulo estudiantes y escoger graduados, luego se listan los datos de los estudiantes en una tabla, si selecciona un estudiante en seguida tendrá el botón visualizar y el botón editar.	
Observación: Al presionar en el botón visualizar tendrá opciones como editar estudiante o adicionar resultado académico disponible.	

Tabla 15. HU-11 Reintegrar Estudiante dado de Baja

Historia de Usuario 11	
Nombre: Reintegrar Estudiante dado d Baja	Iteración Asignada: 1
Prioridad en Negocio: Media	Riesgos de Desarrollo: Alta
Puntos Estimados: 1	Puntos Reales: 1
Descripción: El usuario debe ir al módulo estudiantes y escoger verificar baja, luego se listan los datos de los estudiantes en una tabla, debe selecciona un estudiante o más, en seguida tendrá el botón reintegrar disponible.	
Observación:	

Tabla 18. HU-14 Gestionar Resultado académico

Historia de Usuario 14	
Nombre: Gestionar Resultado académico	Iteración Asignada: 1
Prioridad en Negocio: Baja	Riesgos de Desarrollo: media
Puntos Estimados: 2	Puntos Reales: 2
<p>Descripción: El usuario debe ir al módulo estudiante y escoger licenciatura, postgrado, graduados o verificar baja, luego se listan los datos de los estudiantes en una tabla. Tendrá que seleccionar un estudiante, presionar en visualizar.</p> <p>En visualizar presionar en actualizar resultado y adicionar. Podrá listar los resultados en una pequeña tabla donde tendrá la opción de eliminar o editar los resultados académicos.</p>	
Observación:	

Tabla 19. HU-15 Gestionar País

Historia de Usuario 15	
Nombre: Gestionar País	Iteración Asignada: 3
Prioridad en Negocio: baja	Riesgos de Desarrollo: baja
Puntos Estimados: 2	Puntos Reales: 1
<p>Descripción: El usuario debe ir al módulo configuración y escoger país, luego se listan los datos en una tabla.</p> <p>Puede adicionar nuevo país, para eliminar un país debe selecciona uno o más, en seguida tendrá el botón eliminar disponible. Para que el botón editar aparezca debe seleccionar solo uno país.</p>	
Observación:	

Tabla 20. HU-16 Gestionar Provincia

Historia de Usuario 16

Nombre: Gestionar Provincia	Iteración Asignada: 2
Prioridad en Negocio: baja	Riesgos de Desarrollo: baja
Puntos Estimados: 2	Puntos Reales: 1
<p>Descripción: El usuario debe ir al módulo configuración y escoger provincia, luego se listan los datos en una tabla.</p> <p>Puede adicionar nueva provincia, para eliminar una provincia debe selecciona uno o más, en seguida tendrá el botón eliminar disponible. Para que el botón editar aparezca debe seleccionar solo una provincia.</p> <p>Observación: Para adicionar una provincia debe verificar que el país de la provincia ya se encuentra adicionado en el sistema</p>	

Tabla 22. HU-18 Gestionar Área de Formación

Historia de Usuario 18	
Nombre: Gestionar Área de Formación	Iteración Asignada: 2
Prioridad en Negocio: baja	Riesgos de Desarrollo: baja
Puntos Estimados: 2	Puntos Reales: 1
<p>Descripción: El usuario debe ir al módulo configuración y escoger Área de Formación, luego se listan los datos en una tabla.</p> <p>Puede adicionar nueva Área de Formación, para eliminar una Área de Formación debe seleccionar una o más, en seguida tendrá el botón eliminar disponible. Para que el botón editar aparezca debe seleccionar solo una Área de Formación.</p> <p>Observación:</p>	

Tabla 23. HU-19 Gestionar Carrera

Historia de Usuario 19	
Nombre: Gestionar Carrera	Iteración Asignada: 2

Prioridad en Negocio: baja	Riesgos de Desarrollo: baja
Puntos Estimados: 2	Puntos Reales: 2
<p>Descripción: El usuario debe ir al módulo configuración y escoger Carrera, luego se listan los datos en una tabla.</p> <p>Puede adicionar nueva carrera, para eliminar una carrera debe seleccionar una o más, en seguida tendrá el botón eliminar disponible. Para que el botón editar aparezca debe seleccionar solo una carrera.</p> <p>Observación: Para adicionar una carrera debe verificar que el área de formación de la carrera ya se encuentra adicionado en el sistema</p>	

Anexo 2: Tarjetas CRC.

Tabla 30. Tarjeta CRC Resultado académico

Tarjeta CRC	
Clase: Resultado académico	
Responsabilidad	Colaboración
<p>Contiene toda la información de los resultados académico de un estudiante.</p> <p>Permite adicionar resultado académico de un estudiante, cambiar los atributos del resultado académico de un estudiante.</p> <p>Permite listar y eliminar</p>	Estudiante

Tabla 31. Tarjeta CRC Carrera

Tarjeta CRC	
Clase: Carrera	
Responsabilidad	Colaboración

<p>Contiene toda la información de las carreras de los estudiantes.</p> <p>Permite adicionar una carrera de los estudiantes según el área de formación, cambiar los atributos de las carreras de los estudiantes.</p> <p>Permite listar y eliminar</p>	<p>Área de formación</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------

Tabla 32. Tarjeta CRC Universidad

Tarjeta CRC	
Clase: Universidad	
Responsabilidad	Colaboración
<p>Contiene toda la información de las universidades de los estudiantes.</p> <p>Permite adicionar universidad según la provincia, cambiar los atributos de la universidad.</p> <p>Permite listar y eliminar</p>	<p>Provincia</p>

Tabla 33. Tarjeta CRC Provincia

Tarjeta CRC	
Clase: Provincia	
Responsabilidad	Colaboración
<p>Contiene toda la información de las provincias de los estudiantes.</p> <p>Permite adicionar una provincia según el país, cambiar los atributos de la provincia.</p>	<p>País</p>

Permite listar y eliminar	
---------------------------	--

Tabla 34. Tarjeta CRC Área de Formación

Tarjeta CRC	
Clase: Área de Formación	
Responsabilidad	Colaboración
<p>Contiene toda la información del área de formación de las carreras.</p> <p>Permite adicionar un área de formación, cambiar los atributos del área de formación.</p> <p>Permite listar y eliminar</p>	

Tabla 35. Tarjeta CRC País

Tarjeta CRC	
Clase: País	
Responsabilidad	Colaboración
<p>Contiene toda la información del país de las provincias.</p> <p>Permite adicionar un país, cambiar los atributos del país.</p> <p>Permite listar y eliminar</p>	

Tabla 36. Tarjeta CRC Motivo de Baja

Tarjeta CRC

Clase: Motivo de Baja	
Responsabilidad	Colaboración
<p>Contiene toda la información del motivo de baja de las bajas académicas.</p> <p>Permite adicionar un motivo de baja, cambiar los atributos del motivo de baja.</p> <p>Permite listar y eliminar</p>	

Tabla 37. Tarjeta CRC Usuario

Tarjeta CRC	
Clase: Usuario	
Responsabilidad	Colaboración
<p>Contiene toda la información del usuario del sistema.</p> <p>Permite adicionar un usuario, cambiar los atributos del usuario y autenticar un usuario.</p> <p>Permite listar y eliminar</p>	

Anexo 3: Tarea d Ingeniería.

Tabla 39. Tarea de ingeniería Desarrollar la base de la estructura de la interface de usuario

Tarea de ingeniería	
No. tarea: 2	Historia de Usuario 1: Adicionar Estudiante
Nombre tarea: Desarrollar la base de la estructura de la interface de usuario	
Tipo de tarea: Desarrollo	Punto estimado: 1

Programador responsable: Pedro M. Nekaka
Descripción: Desarrollar la base de la estructura de la interface de usuario en react.js

Tabla 41. Tarea de ingeniería listar estudiante

Tarea de ingeniería	
No. tarea: 4	Historia de Usuario 2: Listar Estudiante
Nombre tarea: Listar Estudiante	
Tipo de tarea: Desarrollo	Punto estimado: 1
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir listar los estudiantes	

Tabla 42. Tarea de ingeniería ver estudiante

Tarea de ingeniería	
No. tarea: 5	Historia de Usuario 3: Ver Estudiante
Nombre tarea: Ver Estudiante	
Tipo de tarea: Desarrollo	Punto estimado: 1
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir ver el registro de un estudiante	

Tabla 43. Tarea de ingeniería dar de baja a un estudiante

Tarea de ingeniería	
No. tarea: 6	Historia de Usuario 4: Dar de Baja a un Estudiante
Nombre tarea: Dar de Baja a un Estudiante	

Tipo de tarea: Desarrollo	Punto estimado: 1
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir dar de baja a un estudiante	

Tabla 44. Tarea de ingeniería buscar por nombre

Tarea de ingeniería	
No. tarea: 7	Historia de Usuario 5: Buscar Por Nombre
Nombre tarea: Buscar Por Nombre Buscar Por Nombre	
Tipo de tarea: Desarrollo	Punto estimado: 1
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir Buscar Por Nombre, mientras el usuario escribe el nombre los resultados debe ser evidentes ya en la tabla	

Tabla 45. Tarea de ingeniería graduar estudiante

Tarea de ingeniería	
No. tarea: 8	Historia de Usuario 6: Graduar Estudiante
Nombre tarea: Graduar Estudiante	
Tipo de tarea: Desarrollo	Punto estimado: 1
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir graduar los estudiantes del último año, o sea cambiar el estado del mismo.	

Tabla 46. Tarea de ingeniería transitar de clase a un estudiante

Tarea de ingeniería	
No. tarea: 9	Historia de Usuario 7: Transitar de Clase a un estudiante
Nombre tarea: Transitar de Clase a un estudiante	
Tipo de tarea: Desarrollo	Punto estimado: 1
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir transitar de clase a un estudiante o mas	

Tabla 47. Tarea de ingeniería editar estudiante

Tarea de ingeniería	
No. tarea: 10	Historia de Usuario 8: Editar Estudiante
Nombre tarea: Editar Estudiante	
Tipo de tarea: Desarrollo	Punto estimado: 2
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir los datos de un estudiante.	

Tabla 48. Tarea de ingeniería listar estudiantes dados de baja

Tarea de ingeniería	
No. tarea: 11	Historia de Usuario 9: Listar estudiantes dado de baja
Nombre tarea: Listar estudiantes dado de baja	
Tipo de tarea: Desarrollo	Punto estimado: 1
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir listar todos los estudiantes que fueron dados de baja.	

Tabla 49. Tarea de ingeniería listar estudiantes graduados

Tarea de ingeniería	
No. tarea: 12	Historia de Usuario 10: Listar estudiantes graduados
Nombre tarea: Listar estudiantes graduados	
Tipo de tarea: Desarrollo	Punto estimado: 1
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir listar todos los estudiantes que ya han terminado sus estudios.	

Tabla 49. Tarea de ingeniería reintegrar estudiante dado de baja

Tarea de ingeniería	
No. tarea: 13	Historia de Usuario 11: Reintegrar Estudiante dado de Baja
Nombre tarea: Reintegrar Estudiante dado de Baja	
Tipo de tarea: Desarrollo	Punto estimado: 1
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir reintegrar todos los estudiantes que fueron dados de baja y que por algún motivo deben reincorporarse en el sistema como estudiantes activos.	

Tabla 50. Tarea de ingeniería gestionar resultado académico

Tarea de ingeniería	
No. tarea: 14	Historia de Usuario 12: Gestionar Resultado Académico
Nombre tarea: Gestionar Resultado Académico	

Tipo de tarea: Desarrollo	Punto estimado: 2
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir adicionar, editar, listar, eliminar resultados académico de un estudiante. Esa interface estará en la interface de visualizar estudiante.	

Tabla 51. Tarea de ingeniería gestionar provincia

Tarea de ingeniería	
No. tarea: 15	Historia de Usuario 13: Gestionar Provincia
Nombre tarea: Gestionar Provincia	
Tipo de tarea: Desarrollo	Punto estimado: 2
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir adicionar, editar, listar, eliminar provincias de un estudiante. Provincias donde están localizados en Cuba y provincias donde vivían en su país de origen.	

Tabla 53. Tarea de ingeniería gestionar área de formación.

Tarea de ingeniería	
No. tarea: 17	Historia de Usuario 15: Gestionar Área de Formación
Nombre tarea: Gestionar Área de Formación	
Tipo de tarea: Desarrollo	Punto estimado: 2
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir adicionar, editar, listar, eliminar el área de formación de sus carrera.	

Tabla 54. Tarea de ingeniería gestionar Carrera

Tarea de ingeniería	
No. tarea: 18	Historia de Usuario 16: Gestionar Carrera
Nombre tarea: Gestionar Carrera	
Tipo de tarea: Desarrollo	Punto estimado: 2
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir adicionar, editar, listar, eliminar las carreras que frecuentan los estudiantes.	

Tabla 55. Tarea de ingeniería gestionar país

Tarea de ingeniería	
No. tarea: 19	Historia de Usuario 17: Gestionar País
Nombre tarea: Gestionar País	
Tipo de tarea: Desarrollo	Punto estimado: 2
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir adicionar, editar, listar, eliminar los países de origen de los estudiantes y el país actual, para facilitar identificar en qué país pertenece cada provincia	

Tabla 57. Tarea de ingeniería gestionar motivo de baja

Tarea de ingeniería	
No. tarea: 21	Historia de Usuario 19: Gestionar Motivo de Baja
Nombre tarea: Gestionar Motivo de Baja	

Tipo de tarea: Desarrollo	Punto estimado: 2
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir adicionar, editar, listar, eliminar los motivos de baja a los estudiantes.	

Tabla 59. Tarea de ingeniería login

Tarea de ingeniería	
No. tarea: 23	Historia de Usuario 20: Login
Nombre tarea: Login	
Tipo de tarea: Desarrollo	Punto estimado: 1
Programador responsable: Pedro M. Nekaka	
Descripción: Desarrollar la interface de usuario para permitir los usuarios del sistema tener acceso al sistema y comprobar que realmente son los que dicen ser. Según el privilegio de acceso podrán manejar el sistema. Desarrollar los métodos necesarios para que haya seguridad en el proceso de verificación del usuario y de la contraseña puesta.	

Tabla 60. Tarea de ingeniería perfil del usuario

Tarea de ingeniería	
No. tarea: 24	Historia de Usuario 20: Perfil de Usuario
Nombre tarea: Perfil de Usuario	
Tipo de tarea: Desarrollo	Punto estimado: 2
Programador responsable: Pedro M. Nekaka	

Descripción: Desarrollar la interface de usuario para permitir a un usuario editar su perfil, cambiar su contraseña principalmente. Además permitir que el usuario pueda hacer un logout, salir del sistema seguramente.

Anexo 4: Prueba de Aceptación.

Tabla 66. Caso de Prueba de aceptación HU6_P6 para la HU “Graduar estudiante”

Caso de Prueba de Aceptación	
Código: HU6_P6	Historia de Usuario 6: Graduar estudiante
Nombre: Graduar estudiante	
Descripción: Prueba la funcionalidad de graduar estudiante.	
Condiciones de ejecución: El usuario debe estar autenticado. Es necesario tener el estudiante registrado en el sistema.	
Pasos de ejecución: El usuario debe ir al módulo estudiantes y escoger entre estudiante de postgrado o de licenciatura, luego se listan los datos de los estudiantes en una tabla, se selecciona el estudiante o los estudiantes en seguida presionar en graduar, botón que se torna visible después que se haga la selección de uno o más estudiantes. En el apartado de visualizar estudiante también la funcionalidad está disponible	
Resultados esperado: El sistema muestra una interface de confirmación, si el usuario confirmar le sale un mensaje de suceso, si no se cancela la operación.	

Evaluación de la prueba: Prueba satisfactoria.

Tabla 67. Caso de Prueba de aceptación HU7_P7 para la HU “Transitar de clase a un estudiante”

Caso de Prueba de Aceptación	
Código: HU7_P7	Historia de Usuario 7: Transitar de clase a un estudiante
Nombre: Transitar de clase a un estudiante	
Descripción: Prueba la funcionalidad de transitar de clase a un estudiante	
Condiciones de ejecución: El usuario debe estar autenticado. Es necesario tener el estudiante registrado en el sistema. Solo transita de clase, o año académico los estudiantes que no son finalistas.	
Pasos de ejecución: El usuario debe ir al módulo estudiantes y escoger entre estudiante de postgrado o de licenciatura, luego se listan los datos de los estudiantes en una tabla, se selecciona el estudiante o los estudiantes en seguida presionar en transitar, botón que se torna visible después que se haga la selección de uno o más estudiantes. Cuando se visualiza un estudiante también está disponible esa opción, solo hay que presionar en ‘Finalizar’ llenar el formulario y presionar en ok.	
Resultados esperado: El sistema muestra un mensaje de suceso, en el caso de que haya error muestra un mensaje del error. Al transitar estudiantes automáticamente el sistema debe también adicionar los resultados académicos, donde los campos media académica y cantidad de arrastre deben ser (-), pero el año académico = al año que cursa, resultado = aprobado, curso = al curso lectivo que culmino.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 68. Caso de Prueba de aceptación HU8_P8 para la HU "Editar estudiante"

Caso de Prueba de Aceptación	
Código: HU8_P8	Historia de Usuario 8: Editar Estudiante
Nombre: Editar Estudiante	
Descripción: Prueba la funcionalidad de editar estudiante	
Condiciones de ejecución: El usuario debe estar autenticado. Es necesario tener el estudiante registrado en el sistema.	
Pasos de ejecución: El usuario debe ir al módulo estudiantes y escoger entre estudiante de postgrado, graduados, verificar baja o de licenciatura, luego se listan los datos de los estudiantes en una tabla, se selecciona el estudiante en seguida presionar en editar, botón que se torna visible solo después que se haga la selección de uno estudiantes. Cuando se visualiza un estudiante también está disponible esa opción, solo hay que presionar en 'Editar' llenar el formulario. Para tal, el usuario debe asegurarse que los datos entrados al sistema sean correctos y evitar dejar campos vacíos. El usuario debe presionar en "Guardar"	
Resultados esperado: El sistema muestra un mensaje de suceso caso todo este bien, caso contrario se muestra un mensaje de error.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 69. Caso de Prueba de aceptación HU9_P9 para la HU "Listar estudiante dado de baja"

Caso de Prueba de Aceptación	
Código: HU9_P9	Historia de Usuario 9: Listar estudiante dado de baja

Nombre: Listar estudiante dado de baja
Descripción: Prueba la funcionalidad de listar estudiante dado de baja
Condiciones de ejecución: El usuario debe estar autenticado. Es necesario tener el estudiante registrado en el sistema.
Pasos de ejecución: El usuario debe ir al módulo estudiantes y escoger verificar baja, luego se listan los datos de los estudiantes en una tabla, si selecciona un estudiante en seguida tendrá el botón visualizar.
Resultados esperado: El sistema busca en la base de datos los estudiantes dado de baja, caso exista se muestra en una tabla caso contrario, muestra una tabla sin registro.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 70. Caso de Prueba de aceptación HU10_P10 para la HU "Listar estudiante graduados"

Caso de Prueba de Aceptación	
Código: HU10_P10	Historia de Usuario 10: Listar estudiante graduados
Nombre: Listar estudiante graduados	
Descripción: Prueba la funcionalidad de listar estudiante graduados	
Condiciones de ejecución: El usuario debe estar autenticado. Es necesario tener el estudiante registrado en el sistema.	
Pasos de ejecución:	

El usuario debe ir al módulo estudiantes y escoger graduados, luego se listan los datos de los estudiantes en una tabla, si selecciona un estudiante en seguida tendrá el botón visualizar y el botón editar.

Resultados esperado:

El sistema busca en la base de datos los estudiantes graduados, caso exista se muestra en una tabla caso contrario, muestra una tabla sin registro.

Evaluación de la prueba: Prueba satisfactoria.

Tabla 71. Caso de Prueba de aceptación HU11_P11 para la HU "Reintegrar estudiante dado de baja"

Caso de Prueba de Aceptación	
Código: HU11_P11	Historia de Usuario 11: Listar estudiante graduados
Nombre: Reintegrar estudiante dado de baja	
Descripción: Prueba la funcionalidad de reintegrar estudiante dado de baja	
Condiciones de ejecución: El usuario debe estar autenticado. Es necesario tener el estudiante registrado en el sistema.	
Pasos de ejecución: El usuario debe ir al módulo estudiantes y escoger verificar baja, luego se listan los datos de los estudiantes en una tabla, debe selecciona un estudiante o más, en seguida tendrá el botón reintegrar disponible.	
Resultados esperado: El sistema muestra un mensaje de confirmación, caso sea positivo el sistema muestra un mensaje de suceso, caso contrario cancela la operación.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 72. Caso de Prueba de aceptación HU12_P12 para la HU "Filtro"

Caso de Prueba de Aceptación	
Código: HU12_P12	Historia de Usuario 12: Filtro
Nombre: Filtro	
Descripción: Prueba la funcionalidad de filtro	
<p>Condiciones de ejecución:</p> <p>El usuario debe estar autenticado.</p> <p>Es necesario tener el estudiante registrado en el sistema.</p>	
<p>Pasos de ejecución:</p> <p>El usuario debe ir al módulo estudiantes y escoger entre estudiante de postgrado, graduados, verificar baja o de licenciatura, luego se listan los datos de los estudiantes en una tabla, bajo la caja de texto, clicando en el botón filtrar por se hace visible todos los filtros. Se puede hacer todas las combinaciones posibles para que la búsqueda sea más eficiente, también la caja de texto funcionara para combinar la búsqueda por nombre con los demás filtros. Se puede filtrar por: provincia, universidad, carrera, área de formación, sexo, año académico.</p> <p>Los filtros son seleccionado al presionar en cada filtro, también puede escribir en los filtros para buscar lo que se desea.</p>	
<p>Resultados esperado:</p> <p>El sistema muestra el resultado de la búsqueda en la tabla caso haya concordancia con lo que se busca, caso contraria se muestra una tabla sin registro.</p>	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 73. Caso de Prueba de aceptación HU13_P13 para la HU "Gestionar usuario"

Caso de Prueba de Aceptación

Código: HU13_P13	Historia de Usuario 13: Gestionar Usuario
Nombre: Gestionar Usuario	
Descripción: Prueba la funcionalidad de gestionar usuario	
Condiciones de ejecución:	
<p>El usuario debe estar autenticado.</p> <p>El usuario debe ser de nivel 0 para poder dar de baja a un estudiante.</p> <p>Es necesario tener el estudiante registrado en el sistema.</p>	
Pasos de ejecución:	
<p>El usuario debe ir al módulo configuración y escoger usuario, luego se listan los datos de usuarios en una tabla. Se podrá ver la foto, nombre, usuario, email y el nivel de acceso.</p> <p>Puede adicionar nuevo usuario, para eliminar un usuario debe selecciona uno o más usuarios, en seguida tendrá el botón eliminar disponible. Para que el botón editar aparezca debe seleccionar solo uno usuario.</p> <p>Solo los usuarios de nivel 0 tendrá acceso al apartado usuario. Para editar el usuario solo podrá cambiar el nivel de acceso, los demás datos no.</p> <p>Los demás dados, como contraseña, nombre, email, usuario, imagen, solo puede cambiar el propietario de la misma cuenta en la cabecera del sistema, clicando en un nombre de usuario e en seguida en perfil.</p> <p>Para autenticarse es necesario en la página login, entrar con el nombre de usuario y la contraseña. La contraseña entrada es comparada con el hash en la base de dados, si el usuario es quien dice ser y la contraseña la correcta tendrá acceso al sistema, según el nivel</p> <p>Al adicionar nuevo usuario, puede seleccionar una imagen o no, debe poner una contraseña por defecto y después avisar al usuario para cambiar la contraseña una vez este autenticado</p>	
Resultados esperado:	

El sistema muestra una interface donde se listan los usuario, caso no haya usuarios no se muestra ningún registro en la tabla.

Al seleccionar un usuario y presionar en editar sale un formulario, después de ser llenado y guardado el sistema muestra un mensaje de suceso. El mismo sucede al adicionar un nuevo usuario clicando en el botón correspondiente.

Evaluación de la prueba: Prueba satisfactoria.

Tabla 74. Caso de Prueba de aceptación HU14_P14 para la HU “Gestionar Resultado Académico”

Caso de Prueba de Aceptación	
Código: HU14_P14	Historia de Usuario 14: Gestionar Resultado Académico
Nombre: Gestionar Resultado Académico	
Descripción: Prueba la funcionalidad de gestionar resultado académico	
Condiciones de ejecución: El usuario debe estar autenticado. Es necesario tener el estudiante registrado en el sistema.	
Pasos de ejecución: El usuario debe ir al módulo estudiante y escoger licenciatura, postgrado, graduados o verificar baja, luego se listan los datos de los estudiantes en una tabla. Tendrá que seleccionar un estudiante, presionar en visualizar. En visualizar presionar en actualizar resultado, llenar el formulario y adicionar. Podrá listar los resultados en una pequeña tabla donde tendrá la opción de eliminar o editar los resultados académicos.	
Resultados esperado: El sistema muestra un mensaje de suceso al adicionar el resultado. Se muestra la interface de visualizar estudiante y abajo tendrá la lista de los resultados adicionados.	

Si el usuario eliminar o editar también se muestra un mensaje de suceso.

Evaluación de la prueba: Prueba satisfactoria.

Tabla 75. Caso de Prueba de aceptación HU15_P15 para la HU "Gestionar País"

Caso de Prueba de Aceptación	
Código: HU15_P15	Historia de Usuario 15: Gestionar País
Nombre: Gestionar País	
Descripción: Prueba la funcionalidad de gestionar país	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: El usuario debe ir al módulo configuración y escoger país, luego se listan los datos en una tabla. Puede adicionar nuevo país, para eliminar un país debe selecciona uno o más, en seguida tendrá el botón eliminar disponible. Para que el botón editar aparezca debe seleccionar solo uno país. Debe llenar el formulario requerido y presionar en ok.	
Resultados esperado: El sistema muestra un mensaje de suceso al adicionar, eliminar o editar también. Los resultados se muestran en la tabla	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 76. Caso de Prueba de aceptación HU16_P16 para la HU "Gestionar Provincia"

Caso de Prueba de Aceptación	
Código: HU16_P16	Historia de Usuario 16: Gestionar Provincia

Nombre: Gestionar Provincia
Descripción: Prueba la funcionalidad de gestionar provincia
Condiciones de ejecución: <p>El usuario debe estar autenticado.</p> <p>Para adicionar una provincia debe verificar que el país de la provincia ya se encuentra adicionado en el sistema</p>
Pasos de ejecución: <p>El usuario debe ir al módulo configuración y escoger provincia, luego se listan los datos en una tabla. presionar en adicionar y llenar el formulario puede adicionar una nueva provincia, para eliminar una provincia debe selecciona uno o más, en seguida tendrá el botón eliminar disponible. Para que el botón editar aparezca debe seleccionar solo una provincia.</p>
Resultados esperado: <p>El sistema muestra un mensaje de suceso al adicionar, eliminar o editar también. Los resultados se muestran en la tabla</p>
Evaluación de la prueba: Prueba satisfactoria.

Tabla 77. Caso de Prueba de aceptación HU17_P17 para la HU "Gestionar Universidad"

Caso de Prueba de Aceptación	
Código: HU17_P17	Historia de Usuario 17: Gestionar Universidad
Nombre: Gestionar Universidad	
Descripción: Prueba la funcionalidad de gestionar universidad	
Condiciones de ejecución: <p>El usuario debe estar autenticado.</p>	

Para adicionar una universidad debe verificar que la provincia de la universidad ya se encuentra adicionado en el sistema.

Pasos de ejecución:

El usuario debe ir al módulo configuración y escoger “universidade”, luego se listan los datos en una tabla.

presionar en adicionar y llenar el formulario puede adicionar una nueva universidad, para eliminar universidad debe selecciona uno o más, en seguida tendrá el botón eliminar disponible. Para que el botón editar aparezca debe seleccionar solo una universidad.

Resultados esperado:

El sistema muestra un mensaje de suceso al adicionar, eliminar o editar también. Los resultados se muestran en la tabla

Evaluación de la prueba: Prueba satisfactoria.

Tabla 78. Caso de Prueba de aceptación HU18_P18 para la HU “Gestionar Área de Formación”

Caso de Prueba de Aceptación	
Código: HU18_P18	Historia de Usuario 18: Gestionar Área de Formación
Nombre: Gestionar Área de Formación	
Descripción: Prueba la funcionalidad de gestionar área de formación	
Condiciones de ejecución:	
El usuario debe estar autenticado.	
Pasos de ejecución:	
El usuario debe ir al módulo configuración y escoger “Área de Formação”, luego se listan los datos en una tabla.	

presionar en adicionar y llenar el formulario puede adicionar una nueva área de formación, para eliminar área de formación debe selecciona uno o más, en seguida tendrá el botón eliminar disponible. Para que el botón editar aparezca debe seleccionar solo una área de formación.

Resultados esperado:

El sistema muestra un mensaje de suceso al adicionar, eliminar o editar también. Los resultados se muestran en la tabla

Evaluación de la prueba: Prueba satisfactoria.

Tabla 79. Caso de Prueba de aceptación HU19_P19 para la HU "Gestionar Carrera"

Caso de Prueba de Aceptación	
Código: HU19_P19	Historia de Usuario 19: Gestionar Carrera
Nombre: Gestionar Carrera	
Descripción: Prueba la funcionalidad de gestionar carrera	
Condiciones de ejecución:	
El usuario debe estar autenticado.	
Para adicionar una carrera debe verificar que el área de formación de la carrera ya se encuentra adicionado en el sistema	
Pasos de ejecución:	
El usuario debe ir al módulo configuración y escoger "Curso", luego se listan los datos en una tabla.	
presionar en adicionar y llenar el formulario puede adicionar una nueva carrera, para eliminar carrera debe selecciona uno o más, en seguida tendrá el botón eliminar disponible. Para que el botón editar aparezca debe seleccionar solo una carrera.	
Resultados esperado:	

El sistema muestra un mensaje de suceso al adicionar, eliminar o editar también. Los resultados se muestran en la tabla

Evaluación de la prueba: Prueba satisfactoria.

Tabla 80. Caso de Prueba de aceptación HU20_P20 para la HU “Gestionar Motivo de Baja”

Caso de Prueba de Aceptación	
Código: HU20_P20	Historia de Usuario 20: Gestionar Motivo de Baja
Nombre: Gestionar Motivo de Baja	
Descripción: Prueba la funcionalidad de gestionar motivo de baja	
Condiciones de ejecución: El usuario debe estar autenticado	
Pasos de ejecución: El usuario debe ir al módulo configuración y escoger “Motivo de baixa”, luego se listan los datos en una tabla. presionar en adicionar y llenar el formulario puede adicionar un nuevo motivo de baja, para eliminar motivo de baja debe selecciona uno o más, en seguida tendrá el botón eliminar disponible. Para que el botón editar aparezca debe seleccionar solo un motivo de baja.	
Resultados esperado: El sistema muestra un mensaje de suceso al adicionar, eliminar o editar también. Los resultados se muestran en la tabla	
Evaluación de la prueba: Prueba satisfactoria.	

Anexo 5: Entrevista.

La entrevista al cliente se realizó en un encuentro marcado en la embajada de Angola en Cuba. Las preguntas fluyeron en dependencia del conocimiento del cliente sobre la gestión de los procesos que se realizan en el SAE.

1. ¿Cómo funciona el proceso de asignación de becas a los estudiantes?
2. ¿Cuál es el rol del SAE en todo el proceso?
3. ¿Qué sistema se usa actualmente para la gestión de la información?
4. ¿Qué limitaciones presenta el sistema actual?
5. ¿De qué manera las limitaciones influyen en el buen funcionamiento de la institución?
6. ¿Quiénes son los que tienen acceso al sistema?
7. ¿Cómo se garantiza la seguridad del sistema actual?
8. ¿Cómo se conservan los datos de los estudiantes que no están en activo?
9. ¿Cómo se registra la trayectoria académica de los estudiantes?
10. ¿Qué se hace con los estudiantes dados de baja académica?
11. ¿Según el problema identificado que es lo que se requiere para tener una solución?
12. ¿Cuáles son los requisitos requeridos a la propuesta de la solución?
13. ¿Si se logra una solución cuales son los beneficios de la misma para el SAE, para el INAGBE y a los estudiantes?

Entrevista realizada a:

Director del SAE: Dr. Eugenio Novais