



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

---

**Título:** Desarrollo del módulo de reporte estadístico para el sistema de gestión del Sector de Apoyo Estudiantil.

**Autor:** Eldade Joaquim Bondo

**Tutores:** Ing. José Alejandro García Calderón

Ing. Frank Fernández Escalona

**Co-Tutor:** Ing. Lizandra Pérez Mejías

La Habana, junio 2020



” La educación es el pasaporte hacia el futuro, el mañana pertenece a aquellos que se preparan para él en el día de hoy.”

**Malcolm X**

## **DECLARACIÓN DE AUTORÍA**

Declaro por este medio que yo Eldade Joaquim Bondo, con carné de identidad 94092701006 soy el autor principal del trabajo titulado “Desarrollo del módulo de reporte estadístico para el sistema de gestión del Sector de Apoyo Estudiantil” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmo la presente a los 19 días del mes de junio del año 2020.

Autor:

---

Eldade Joaquim Bondo

Tutores:

---

Ing. José Alejandro García Calderón

---

Ing. Frank Fernández Escalona

Co-tutor:

---

Ing. Lizandra Pérez Mejías

## DATOS DE CONTACTO

### Tutores:

**Ing. José Alejandro García Calderón:** El ingeniero José Alejandro García Calderón graduado en el año 2009 se ha desempeñado como profesor de programación, cumplió misión en la hermana República Popular de Angola participando en el desarrollo de un sistema para el Ministerio de Educación Superior de Angola, actualmente ocupa el cargo de subdirector del Centro de Desarrollo de Tecnologías para la Formación (FORTES).

**Ing. Frank Fernández Escalona** Profesor Instructor. Graduado de Ingeniero en Ciencias Informáticas en el año 2015. Se ha desempeñado como profesor de Introducción a la Programación y Programación 1 durante cinco cursos. Ha participado en Fórum de ciencias y técnica municipal y provincial.

### Co-tutor:

**Ing. Lizandra Pérez Mejías:** Lizandra Pérez Mejías graduada en el año 2015 se ha desempeñado como analista principal y líder de proyecto Intranet en la Dirección de Informatización y actualmente se encuentra desempeñándose como analista del proyecto Akademos en FORTES.

## DEDICATORIA

*A mis padres, mis abuelos y a mis hermanos.*

## **AGRADECIMIENTOS**

*A Dios por el don de la vida, al Instituto Nacional de Becas de Estudio (INAGBE) por la costear mi carrera desde el inicio hasta el fin, al exdirector Dr. Moisés Kafala Neto por su excelente trabajo. También doy gracias al gobierno cubano por aceptar este intercambio de conocimientos, a los directivos de la Universidad de las Ciencias Informáticas por la dedicación y atención que han demostrado durante el periodo que hemos estado aquí.*

*Agradezco a mi cotutora y a mis tutores por la disponibilidad en ayudarme con la corrección del documento, en especial agradezco al Ingeniero José Calderón que además de mi tutor, fue mi profesor y un gran amigo, con él he aprendido mucho sobre la nueva tendencia del desarrollo web. Agradezco al centro FORTES de la Universidad de las Ciencias Informáticas, por la oportunidad de poder allí trabajar, he aprendido mucho durante me instancia allí.*

*También doy gracias al Ingeniero Leonardo Remón que además de ser un gran amigo y compañero en nuestros escenarios musicales, me ha ayudado muchísimo con el documento en la corrección de los errores léxicos y sintácticos del idioma español.*

*Gracias a mi familia por el apoyo moral, a mi hermano Ruben Bondo que me ayudó a enamorarme por las matemáticas, Abner Bondo que me enseña no solo a soñar como también materializarlos, Marinela Bondo que me motiva a estudiar. A mis tres hermanos que la vida me regalo (Sinidio, Josias, João) por reunirse eventualmente para hablar conmigo y motivarme a seguir luchando.*

*Doy gracias a todos los colegas y compañeros que me repasaron cuando no entendía algún contenido, a todos los profesores que con amor y dedicación me enseñaron no solo ciencia como también cosas de la vida.*

*Doy gracias a todos que directa o indirectamente contribuyeron para la realización de este lindo sueño.*

## **RESUMEN**

En el Instituto Nacional de Gestión de Becas de Estudios (en Angola) existe un proyecto que consiste en dar seguimiento al plan de formación de profesionales, que culmina con la ubicación laboral de los estudiantes formados. Para facilitar la distribución de profesionales en las distintas provincias, el Sector de Apoyo Estudiantil (en las embajadas de los países cooperantes), elabora un informe con la cantidad de estudiantes formados cada año, en el mismo constan las especialidades y las provincias de origen. El proceso de elaboración del informe, presenta dificultades en cuanto a la disponibilidad de la información en el período requerido, la precisión de los datos, dificultando el proceso de ubicación laboral. La presente investigación propone el desarrollo de un sistema informático para generar el reporte estadístico de los estudiantes. Se describen además las técnicas para la realización de las pruebas para validar la propuesta de solución.

**Palabras claves:** reporte estadístico, sistema informático, ubicación laboral.

## **ABSTRACT**

*At the National Institute for Study Scholarship Management (in Angola) there is a project that consists of following up on the professional training plan, which culminates in the job placement of the graduated students. To facilitate the distribution of professionals in the different provinces, the Student Support Sector (in the embassies of the cooperating countries), prepares a report with the number of graduated students each year, in which include the specialties and the provinces of origin. The process of preparing the report presents difficulties in terms of the availability of information in the required period, the precision of the data, making the employment a difficult process. The present investigation proposes the development of a computer system to generate the statistical report of the students. The techniques for carrying out the tests to validate the solution proposal are also described.*

**Key words:** *statistical report, computer system, job placement.*

# ÍNDICE

<b>INTRODUCCIÓN</b> .....	14
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA</b> .....	19
1.1. El plan de formación de profesionales.....	19
1.2. Conceptos asociados a la investigación .....	22
1.3. Estado del Arte.....	23
1.4. Tecnologías, lenguaje de programación y herramientas .....	27
1.5. Lenguaje de programación.....	31
1.6. Herramientas a utilizar .....	32
1.7. Metodología de Desarrollo de Software.....	33
<b>CAPÍTULO 2: PROPUESTA DE SOLUCIÓN</b> .....	36
2.1. Descripción de la propuesta de Solución.....	36
2.2. Usuarios del Sistema .....	37
2.3. Lista de reserva del producto .....	38
2.4. Historias de Usuario (HU).....	39
2.5. Requisitos no funcionales .....	42
2.6. Planeación .....	43
2.6.1. Estimación de Esfuerzo por Historias de Usuario .....	43
2.6.2. Plan e Iteraciones para entregables.....	45
2.7. Diseño de la Aplicación .....	48
2.7.1. Tarjetas CRC .....	48
2.8. Arquitectura.....	52
2.9. Patrones de Diseño.....	54

2.10- Conclusiones del capítulo.....	56
<b>CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA .....</b>	<b>57</b>
3.1. Modelo de Datos .....	57
3.2. Tareas de Ingeniería .....	59
3.3. Pruebas.....	63
3.3.1 Pruebas Unitarias .....	63
3.3.2 Pruebas de Integración.....	67
3.3.3 Pruebas de Aceptación.....	71
3.3.4 Resultado de las pruebas realizadas .....	73
3.4. Implementación del sistema .....	74
3.5. Conclusiones del capítulo .....	76
<b>CONCLUSIONES GENERALES .....</b>	<b>77</b>
<b>RECOMENDACIONES.....</b>	<b>78</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>79</b>
<b>ANEXOS.....</b>	<b>83</b>

## ÍNDICE DE TABLAS

Tabla 1. Universidades Públicas de Angola .....	20
Tabla 2. Representación de características convergentes entre los sistemas homólogos.....	26
Tabla 3. Comparación entre las tres tecnologías de desarrollo de vistas del usuario .....	29
Tabla 4. Lista de reserva del producto .....	38
Tabla 5. HU-1 Generar datos estadísticos de pregrado por carrera y género.....	40
Tabla 6. HU-13 Listar estudiantes de pregrado .....	40

Tabla 7. HU-15 Filtrar estudiantes de pregrado.....	40
Tabla 8. HU-35 Generar gráfico de proyección de graduación general .....	41
Tabla 9. HU-38 Generar gráfico de estudiantes de finalistas por provincia .....	41
Tabla 10. HU-20 Exportar estudiantes finalistas.....	42
Tabla 11. Listado de requisitos no funcionales del sistema .....	42
Tabla 12. Estimación de Esfuerzo por HU.....	43
Tabla 13. Estimación de esfuerzo por HU calculados .....	45
Tabla 14. Tarjeta CRC Estudiante.....	48
Tabla 15. Tarjeta CRC Graduado.....	49
Tabla 16. Tarjeta CRC Baja .....	49
Tabla 17. Tarjeta CRC Resultado académico .....	49
Tabla 18. Tarjeta CRC Carrera .....	50
Tabla 19. Tarjeta CRC Universidad.....	50
Tabla 20. Tarjeta CRC Provincia.....	50
Tabla 21. Tarjeta CRC Área de Formación .....	51
Tabla 22. Tarjeta CRC País .....	51
Tabla 23. Tarjeta CRC Motivo de Baja .....	51
Tabla 24. Tarjeta CRC Usuario .....	52
<i>Tabla 25. Generar proyección de graduación.....</i>	<i>59</i>
Tabla 26. Listar estudiantes de pregrado activos. ....	60
Tabla 27. Filtrar estudiantes de pregrado.....	60
Tabla 28. Estadística de estudiantes finalistas .....	61
Tabla 29. Exportar tabla de estudiantes finalistas por provincia y carrera .....	61

Tabla 30. Estadísticas de los estudiantes de postgrado por provincia.....	62
Tabla 31. Gráfico de los estudiantes de pregrado por año. ....	63
Tabla 32. Los casos de prueba realizados para el camino básico.....	66
Tabla 33. Caso de prueba de integración del componente "Reporte de los estudiantes de pregrado". ....	70
Tabla 34. Caso de Prueba de Aceptación HU35_P1 para la HU "Generar gráfico de predicción de graduación".....	71
Tabla 35. Caso de Prueba de Aceptación de HU15_P2 para la HU "Filtrar estudiantes de pregrado".....	71
Tabla 36. Caso de Prueba de Aceptación de HU14_P3 para la HU "Exportar estudiantes de pregrado". ..	72
Tabla 37. Caso de Prueba de Aceptación de HU36_P4 para la HU "Generar gráfico de estudiantes por provincia". ....	72

## ÍNDICE DE FIGURAS

Figura 1. Flujo de información entre las instituciones. ....	22
Figura 2. Resultado de la encuesta comparativa entre marcos de trabajo para interfaces. ....	30
Figura 3. Tendencia de búsqueda en Google Trends.....	31
Figura 4. Listado de los estudiantes.....	37
Figura 5. Diseño conceptual de la arquitectura .....	53
Figura 6. Evidencia de la arquitectura .....	53
Figura 7. Evidencia de la arquitectura. ....	54
Figura 8. Ejemplo del uso de herencia .....	55
Figura 9. Evidencia de la arquitectura. ....	56
Figura 10. Evidencia del modelo. ....	58
Figura 11. Ejemplo de relación entre documentos en MongoDB.....	58
Figura 12. Fragmento de código de búsqueda por nombre.....	65
Figura 13. Representación del grafo de flujo de camino básico.....	65

Figura 14. Resultado de las pruebas por iteración. ....	73
Figura 15. Página principal de la aplicación .....	74
Figura 16. Página de reportes de estudiantes de pregrado. ....	75
Figura 17. Página de estadísticas por gráficos. ....	76
Figura 18. Declaración de compromiso del estudiante becario. ....	84

## INTRODUCCIÓN

La sociedad en diferentes momentos de la historia ha puesto en evidencia la importancia de la educación como factor determinante en el desarrollo y transformación de estas. En este contexto, los docentes han tenido un rol protagónico, aunque la formación permanente e integral fue descuidada por mucho tiempo. A partir del siglo XX comienza a tomar fuerza la necesidad de la formación continua, siendo la formación docente uno de los centros de atención principales desde inicios del siglo XXI. La educación con calidad se orienta a la formación de profesionales integrales, se refiere no solo al aprendizaje de habilidades específicas, sino al desarrollo de las cualidades, valores personales, identidad profesional, social y cultural del contexto particular, insertado en la realidad mundial. [18]

En este sentido los gobiernos de los países juegan un papel fundamental en la gestión de la adecuada formación de profesionales que contribuyan con sus conocimientos al desarrollo de cada país. El gobierno de Angola en su plan nacional de formación de profesionales y como estrategia del Ministerio de Educación Superior, se decide a crear el Instituto Nacional de Gestión de Becas de Estudios (INAGBE). Institución encargada de gestionar las becas de estudio a partir de un conjunto de especialidades con déficit en la nación. Estas becas se clasifican en internas para los candidatos matriculados en las universidades nacionales y externas para las universidades foráneas. Este subsidio es fundamentalmente para estudiantes que soliciten ayuda financiera para cubrir gastos referentes a su formación, tanto de pregrado como de postgrado. Durante el proceso de selección y asignación de becas, se revisan los resultados obtenidos en las enseñanzas precedentes, se comprueba que el candidato cumple los requisitos docentes, se analiza la condición financiera del solicitante, pues la ayuda es principalmente para personas con dificultades económicas. Una vez otorgada la beca, los becarios deben firmar un compromiso que refleja las siguientes pautas: tener un comportamiento moral y cívico irreprochable; buenos resultados académicos; no abandonar la carrera antes del final previsto. En caso de ser becario externo se le añade que este debe regresar al país para contribuir con el desarrollo del mismo. Al incumplir el reglamento, se compromete la beca al punto del cierre del contrato de la misma y en casos excepcionales se le puede solicitar al becario la devolución del presupuesto invertido por el INAGBE hasta el momento.

Como representante de INAGBE en el exterior, surge el Sector de Apoyo a los Estudiantes (SAE), que es el organismo competente del seguimiento y atención de los estudiantes que cursan actividades de pregrado y

postgrado en universidades fuera de Angola. Generalmente radica en las embajadas de Angola en los países que tienen presencia de estudiantes angoleños. Cuba no queda exenta de ser de referencia internacional en el área de la educación superior de pregrado y postgrado, lo que hace posible que a estudiantes de la hermana República de Angola les sea gestionada una beca de estudio en universidades cubanas. Esto tributa a que INAGBE necesite un SAE en territorio cubano. Un estudio realizado en el Sector de Apoyo a los Estudiantes en Cuba, muestra la relación que existe entre esta entidad y las instituciones formadoras del país. El SAE contacta con el Ministerio de Educación Superior (MES) para solicitar plazas en las universidades que posteriormente serán cubiertas con estudiantes angoleños, dentro de sus funciones deberá proceder al pago anual de las carreras cursadas por los estudiantes que logren la beca, contacta con el Ministerio de Salud Pública (MINSAP) para gestionar lo referente a los seguros de salud de cada individuo. Una vez terminado todos los trámites pertinentes y el arribo de los becarios a las universidades, estas comienzan un proceso de comunicación continua con el SAE. Cada universidad deberá tributar un informe con la conducta y los resultados académicos obtenidos por los estudiantes angoleños al final de cada semestre.

Uno de los procesos principales que se desarrolla en el SAE es el pago de los valores correspondientes a cada becario mientras esté vigente el contrato entre el beneficiario e INAGBE. Para que este acuerdo legal mantenga vigencia es requisito promover académicamente y mantener una buena conducta cívica y moral. Con el objetivo de tener una mayor claridad en la organización, cumplimiento de sus funciones y responsabilidades para con los estudiantes, el SAE tributa la información procesada directamente al INAGBE y al Ministerio de Educación Superior de Angola, que son los principales responsables en velar por el presupuesto invertido.

Como resultado del flujo de información entre el SAE y las instituciones cubanas, se crea una base de datos Access donde se almacena la información referente a los estudiantes, actualmente muy limitada, esto provoca que muchos de los reportes son generados manualmente por los especialistas del sector en Excel y documentos Word. El uso del sistema de base de datos Access ocasiona limitaciones en el almacenamiento de la información, para no sobrecargarla, son borrados del sistema los estudiantes que se gradúan o causan baja académica, lo que dificulta la gestión histórica de dicha información. La existencia de tablas no normalizadas en el sistema actual, ha ocasionado dificultad en la recuperación y actualización de datos. No se registra la trayectoria académica de los estudiantes, sólo se conserva la información del año anterior, de forma que cada vez que se procese la información nueva de un semestre, se elimina todo

lo referente a los períodos precedentes. Esta situación dificulta la elaboración de reportes necesarios, complejiza la realización de un trámite de baja académica, entorpece el análisis para proporcionar una oportunidad de repitencia y hace difícil el procesamiento de los pagos a los estudiantes. El sistema actual no cuenta con funcionalidades que automaticen y brinden la información de forma eficiente sobre los procesos que se desarrollan en el SAE.

Analizando lo antes descrito se plantea como **problema científico**: Las deficiencias en la elaboración de reportes estadísticos, ocasionan imprecisiones de los datos y afecta la disponibilidad de la información en el período requerido.

El problema identificado enmarca como **objeto de estudio** el proceso de elaboración de reportes estadísticos con la información de los estudiantes. Se identifica como **campo de acción** los métodos de elaboración de reportes estadísticos para la toma de decisiones en la ubicación laboral.

Para dar solución al problema descrito se propone el siguiente **objetivo general**: Desarrollar una aplicación que permita a los usuarios generar reportes estadísticos de interés para el SAE.

En aras de concretar el objetivo general planteado se han trazado varios **objetivos específicos**:

1. Seleccionar la metodología, herramientas y tecnologías a utilizar en el desarrollo del módulo de reporte estadístico para el sistema de gestión del Sector de Apoyo Estudiantil.
2. Realizar el análisis, diseño y desarrollo del módulo de reporte estadístico para el sistema de gestión del Sector de Apoyo Estudiantil.
3. Realizar pruebas al módulo de reporte estadístico para el sistema de gestión del Sector de Apoyo Estudiantil.

Se tiene como **idea a defender** de la presente investigación: El desarrollo de una aplicación que permita a los usuarios generar reportes estadísticos, reducirá las deficiencias existentes en este proceso, habrá precisión de los datos y la información estará disponible a cualquier momento.

Se utilizaron los siguientes **métodos teóricos**:

- **Histórico – lógico**: Para determinar los antecedentes, tendencias, y regularidades del proceso de generación de reportes estadísticos estudiantiles, sustentado en las tecnologías y las herramientas

informáticas para la gestión de la información.

- **Analítico – sintético:** Se utiliza este método para descomponer el problema de la investigación en elementos atómicos y estudiar de forma detallada cada uno de ellos, para posteriormente integrarlos en la propuesta de solución.
- **Inductivo – deductivo:** este método se utiliza para concluir nuevos conocimientos y predicciones referentes al proceso de generación de reportes estadísticos estudiantiles sustentado en las tecnologías, que en la etapa final son sometidos a un proceso de verificaciones empíricas.

Se emplean los siguientes **métodos empíricos**:

- **Observación:** Se utilizó para obtener de forma directa, la información de la realidad objetiva del comportamiento del proceso de elaboración de reportes estadísticos sustentado en las tecnologías.
- **Entrevistas:** Se realizaron para identificar el grado de conocimiento de los involucrados en el proceso de elaboración de reportes estadísticos, también para una mejor comprensión de la estructura organizativa.
- **Análisis documental:** Se utilizó para analizar, clasificar, verificar y seleccionar de forma sensata los contenidos de las bibliografías, referentes al proceso de elaboración de reportes estadísticos sustentado en las tecnologías. [15]

### **Estructura organizativa del documento**

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

**Capítulo 1: “Fundamentación Teórica”,** se muestran los conceptos fundamentales asociados a la presente investigación para una mejor comprensión del documento, se especifican los conceptos relacionados con la generación de reportes estadísticos, esto permitirá comprender el dominio del problema. Se refleja el estado del arte sobre el tema de investigación teniendo en cuenta una perspectiva nacional, así como internacional. Finalmente se realiza un análisis de las principales tecnologías, arquitectura, metodologías, lenguajes de programación, marco de trabajo y las herramientas que serán utilizadas para el desarrollo de la solución propuesta.

**Capítulo 2: “Propuesta de solución”,** se describe la propuesta de solución a partir de los elementos que se definen en la fase de Exploración y Planificación. Se detallan las Historias de Usuario que recogen todas

las funcionalidades a implementar. Se muestran otros artefactos propios de la metodología de desarrollo seleccionada.

**Capítulo 3: “Implementación y Pruebas**, se definen las pautas de la implementación, así como el desarrollo de pruebas y validaciones del sistema, lo que permitirá establecer estándares de calidad para con los requisitos y funcionalidades que debe suplir la solución propuesta.

## **CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA**

En el presente capítulo se definen un conjunto de conceptos y elementos teóricos que propiciarán un mejor entendimiento de la investigación. Se definen elementos fundamentales sobre el plan de formación de profesionales en Angola. Cuenta con un estudio de sistemas homólogos, donde se analizan las características de sistemas o procesos automatizados que gestionan la información de estudiantes en centros de estudios. Concluyendo con un análisis de las herramientas, tecnologías, lenguajes y metodología de desarrollo de software que serán utilizadas en el desarrollo de la solución fundamentada.

### **1.1. El plan de formación de profesionales**

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), mediante el decreto ley 44530, del 21 de agosto, de la administración portuguesa, que más tarde evolucionó y adquirió el estatus universitario en virtud del decreto ley 48/90, del 23 de diciembre de 1968, que adoptó la designación de Universidad de Luanda. [19]

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 Huíla. 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 pudiera libremente ejercer sus estudios de pregrado. [19] 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 cada año un número significativo de profesionales. [20] 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. Es importante resaltar la posibilidad de que existan otras universidades que no serán mencionadas, tal hecho se justifica debido a que en el portal del Ministerio de Educación Superior no se dispone de dicha información, lo obtenido hasta entonces proviene de tesis investigación. [19]

Tabla 1. Universidades Públicas de Angola

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
<b>VI</b>	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
<b>VII</b>	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
<b>VIII</b>	29	Universidade Cuíto Cuanavale	2009	Cunene, Cuando Cubango

Como el objetivo es crecer y mejorar la calidad de la educación, el país cuenta con instituciones privadas las cuales tienen que cumplir determinados requisitos y estar debidamente registradas en el Ministerio de Educación Superior de Angola, para ser reconocidas legalmente como instituciones de educación superior. Con el creciente desarrollo de la educación y el interés del estado angoleño en la superación universitaria, se creó el INAGBE para el financiamiento de becas para los estudiantes con pocos recursos y excelentes resultados académicos. Esta institución gestiona las becas internas para las universidades en Angola y externas para instituciones en el extranjero. Como representante de la misma en los respectivos países se crea el SAE, que se encarga del seguimiento de los estudiantes y reporta los resultados académicos directamente al INAGBE.

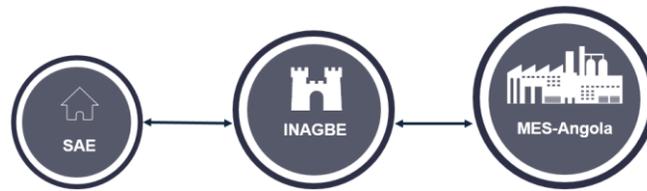


Figura 1. Flujo de información entre las instituciones.

## 1.2. Conceptos asociados a la investigación

La DRAE define **estadística** como estudio de los datos cuantitativos de una población, de los recursos naturales e industriales, del tráfico o de cualquier otra manifestación de las sociedades humanas. Rama de la matemática que utiliza grandes conjuntos de datos numéricos para obtener inferencias basadas en el cálculo de probabilidades.

La **estadística** es la ciencia cuyo objetivo es reunir información cuantitativa concerniente a individuos, grupos, series de hechos, para deducir de ellos, gracias al análisis de los datos, significados precisos o previsiones para el futuro. La estadística, en general, es la ciencia que trata de la recopilación, organización presentación, análisis e interpretación de datos numéricos con el fin de tomar decisiones efectivas y pertinentes. [14]

Luego de analizadas estas definiciones el autor asume que la estadística consiste en la recolección, estudio y análisis de datos referentes a una determinada población que incluye operaciones como extracción, organización y análisis con la finalidad de obtener deducciones que permitan posteriormente la toma de decisiones oportunas y acertadas.

Según la DRAE **reporte** es noticia o informe. También abordan que reportar es transmitir, comunicar o dar noticia.

Analizando lo descrito anteriormente, el autor plantea que **reporte estadístico** es un informe con datos cuantitativos referente a una determinada población.

**Datos persistentes** son aquellos que se mantienen en el tiempo, después de la ejecución del proceso que

los creó. Los datos persistentes se almacenan en una base de datos o archivos que pueden ser leídos o modificados por otros procesos en un momento posterior. [37]

### 1.3. Estado del Arte

Con el propósito de caracterizar sistemas o procesos automatizados que gestionan la información de estudiantes en centros de estudios en el **contexto internacional**, se evidencia que:

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 el método de investigación basado en entrevista realizado a los desarrolladores del Sistema de Gestión de Becas de Estudio (SGBE), se obtuvieron las siguientes informaciones relacionadas a las funcionalidades:

- 1- Procesamientos de candidaturas y atribución de becas. En el proceso de candidatura, se registran los estudiantes que cumplan con los requisitos dictados por el INAGBE. La atribución de becas es realizada por un algoritmo que determina lo siguiente: Seleccionar los estudiantes cuya edad se acerca al 17, que su promedio general se acerca al 20, que estén cursando sus estudios en instituciones públicas y en una rama de la ciencia priorizada para Angola.
- 2- Pagos de los estudiantes y funcionarios del SAE, se realiza a partir de las legislaciones vigentes y reglamentos de las becas de estudios, Estas establecen las cuotas y la frecuencia a pagar a cada involucrado.
- 3- Proceso de renovación de becas. Este proceso se realiza para constatar el aprovechamiento académico y el comportamiento cívico y moral del becario con el objetivo dar continuidad o no al subsidio de la beca obtenida previamente durante otro año.

El SGBE se resume en los procesos antes descritos, lo cual comparte ventajas que sirven de referencia para la presente investigación:

- Generación de reporte de los estudiantes por provincia;
- Generación de reporte de los estudiantes por país;
- Actualización de los datos de los estudiantes;

- Imprimir el listado de los estudiantes;

Para caracterizar sistemas o procesos automatizados que gestionan la información de estudiantes en centros de estudios en el **contexto nacional**, se estudian tres sistemas:

### **Sistema de Gestión Estudiantil (SGE):**

Usando los métodos empíricos basados en análisis documental y entrevistas a los especialistas del SAE-CUBA, se obtuvo la información referente al proceso de gestión estudiantil gestionada en el sistema. 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.
- Reporte estático en formato PDF.
- 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 ayuda para otros sistemas de base de datos, la funcionalidad de reporte generó la idea al autor para automatizar el proceso que se hace de forma manual, pensando entonces en el rediseño e implementación del sistema, usando tecnologías modernas, para que no haya limitaciones en el almacenamiento como es el caso de las bases de datos construidas en Access, añadirle la característica de multiplataforma y un diseño visual adaptativo, intuitivo y moderno.

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 y necesita tiempos de respuestas mínimos. Se hace necesario un sistema que sea capaz de gestionar y brindar la información de forma eficiente, garantizando la seguridad, confiabilidad y disponibilidad de la información.

### **Sistema de Gestión Universitaria:**

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, Laboratorios, Producción, Investigación, Cooperación y Postgrado. Los módulos anteriores se controlan mediante un sistema de sesiones que garantizan el acceso eficiente a la información. [15]

A continuación, se describe el módulo Estudiante del subsistema de 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.

Con lo antes descrito se considera que el SGU comparte ventajas que pueden ser referencias para otros sistemas con características similares. [17]

#### **Directorios de la Universidad de las Ciencias Informáticas:**

Como parte de la comunicación universitaria la UCI cuenta con un directorio institucional, un portal desarrollado por la Dirección de Informatización de la universidad. La parte visible al público, cuenta con los siguientes módulos: Personas, Teléfonos, Aplicaciones y Cumpleaños.

- Personas: En este módulo se realizan búsquedas basadas en filtros por nombres, apellidos, categoría, provincia, género y otras combinaciones.
- Teléfono: Se realizan filtros por área (devuelve el teléfono), por teléfono (devuelve el área).
- Aplicaciones: Se muestra un listado con todos los sistemas desarrollados en la universidad que hayan sido registrado previamente con la Dirección de Informatización.
- Cumpleaños: Se visualiza en orden alfabético todos los cumpleaños del día de las personas que tengan pública dicha información en su perfil.

Los métodos de búsqueda implementados en el directorio pueden servir de referencia para otros sistemas. La visualización y comportamiento de los filtros, enriquecerá la usabilidad de sistemas que generan reportes como lo es el desarrollado en la presente investigación.

Para mejor percepción de las características particulares, obtenidas de los sistemas antes descritos, se denota lo siguiente:

**Información detallada de los estudiantes (C1):** Muestra informaciones de los estudiantes, lo que permite identificar a cada uno.

**Proyección de graduación (C2):** A partir de una población de estudiantes, se tiene en cuenta los años de frecuencia de cada uno, así se proyecta la cantidad de graduados en los próximos seis años.

**Multiplataforma (C3):** Aplicaciones informáticas que se ejecutan en cualquier sistema operativo.

**Aplicación de una sola página (C4):** *Single page Application (SPA)* es una aplicación web que mejora la experiencia del usuario al trabajar dentro de un navegador y no requiere la recarga de la página durante las tareas de trabajo.

Se seleccionan estas características porque se considera necesario para dar cumplimiento a los requerimientos del software. Los sistemas estudiados fueron provechosos, pues sirvieron de referencias para la arquitectura de la información, construcción de la base de datos y características funcionales. A continuación, se muestra la tabla 2 que representa las características convergentes entre los sistemas homólogos.

*Tabla 2. Representación de características convergentes entre los sistemas homólogos*

<b>Aplicaciones / Características</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>
<b>SGBE</b>	Si	No	Si	No
<b>SGU</b>	Si	No	Si	No
<b>Directorio</b>	Si	No	Si	No
<b>SGE</b>	Si	No	No	No

De los sistemas informáticos identificados con el objetivo de gestionar la información de estudiantes y generar reportes estadísticos, no se pudo seleccionar ninguno para solucionar la problemática planteada porque no se adecuan al negocio que se manejan en el SAE, además el SGE al no satisfacer con las

características C2, C3 y C4 y los demás sistemas con las características C2 y C4. Todos estos elementos constituyen un factor importante para decidir apostar en una nueva propuesta de implementación.

Para facilitar el proceso de implementación del nuevo sistema informático, se hace necesario rediseñar la base de datos actual, pues se identificaron tablas no normalizadas. Existen muchos procesos no automatizados lo que afecta la disponibilidad de la información.

## 1.4. Tecnologías, lenguaje de programación y herramientas

### ✓ MERN Stack

Es un entorno de desarrollo compuesto por 4 marcos independientes, Mongo DB, Express JS, React JS y Node JS, de ahí la abreviatura MERN. [3]

**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 programación en la nube. [1]

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. [2]

**Express JS:** Es un marco de Node.js es de código abierto muy ligero, extremadamente sencillo 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 de trabajo, 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

tributando a un desarrollo mucho más rápido y la simple comunicación con complementos de terceros lo convierte en el marco Node.js más popular entre los desarrolladores. [3]

**React JS:** Es una biblioteca de JavaScript de código abierto para construir interfaces de usuario. [4] Es basada en componentes, actualmente es una de las más utilizadas para el desarrollo del lado del cliente, proporciona su comportamiento de forma natural para aplicaciones de una sola página. Fue desarrollado inicialmente por el ingeniero de Facebook llamado Jordan Walke y se actualiza constantemente por esta empresa y la comunidad de desarrollo. Se implementó por primera vez en la fuente de noticias de Facebook en 2011. [3]

**Node Js:** Es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. [5] 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. [3]

## **MEAN Stack**

Es un conjunto de softwares usados 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. [6]

**Angular JS:** Es un marco de trabajo 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. [7]

## 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. [8]

**Vue.js:** es un marco de trabajo progresivo para construir interfaces 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. [8]

Para la selección de la tecnología de desarrollo a utilizar en la investigación, se hizo un análisis entre las tecnologías antes mencionadas. Por lo tanto, el criterio de selección se basa en los tres competidores principales de la actualidad (Angular JS, React JS y Vue.JS). A continuación, se muestra un conjunto de informaciones comparativas que facilitaron la selección de React JS para la implementación. [9]

Analizando la tabla 3 el autor se fija en React JS por tener una curva de aprendizaje media, de ser una librería tiene una mayor flexibilidad, utiliza *JSX* como lenguaje de programación que es una adaptación de código *JavaScript* dentro del *HTML*. Posee un rendimiento que oscila entre el medio y alto, eso se adapta fielmente en las características del sistema que se desea desarrollar. No se escoge Angular por tener una curva de aprendizaje alta y una flexibilidad baja. De igual modo no se selecciona Vue.JS por su bajo rendimiento en comparación con las otras dos tecnologías.

*Tabla 3. Comparación entre las tres tecnologías de desarrollo de vistas del usuario*

Características					
Tecnología	Curva de aprendizaje	Flexibilidad	Tipo	Lenguaje	Rendimiento
Angular	Alta	Baja	Framework	TypeScript	Alto
React JS	Media	Alta	Librería	JSX /JavaScript	Alto/Medio

Vue.JS	Baja	Alta	Framework	HTML /JavaScript	Medio/Bajo
--------	------	------	-----------	------------------	------------

Una encuesta realizada en la comunidad de desarrolladores donde las respuestas que se buscaban eran las siguientes:

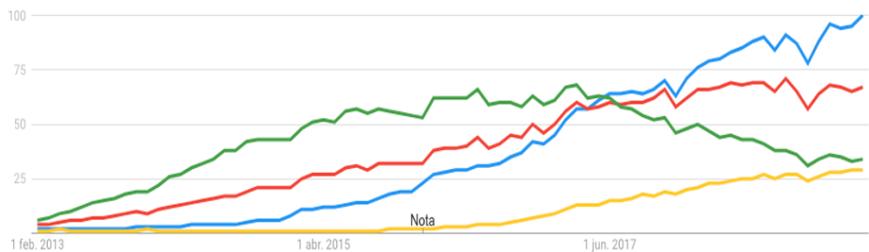
- He usado y volvería a usar;
- He usado y no volvería a usar;
- He oído hablar y me gustaría aprender;
- He oído hablar y no me interesa;
- No he oído hablar.

Esta encuesta refuerza una vez más la decisión de seleccionar React JS, pues la cantidad de personas que han usado y volverían a usar es superior de las demás, eso refleja de cierta forma una gran satisfacción por parte de los usuarios de la tecnología. La figura 2 muestra el resultado de la encuesta.



Figura 2. Resultado de la encuesta comparativa entre marcos de trabajo para interfaces.

Otro elemento que se tuvo en cuenta fue la tendencia de búsqueda en google, lo cual evidencia que React JS es el más buscado en los últimos años.



React	100%
Angular	67%
AngularJS	34%
VUE	29%

Figura 3. Tendencia de búsqueda en Google Trends.

Analizando lo anteriormente descrito, el autor decide asumir el entorno de desarrollo con las siguientes características:

- ✓ Flexible que sea posible la estructuración del proyecto según filosofías propias de trabajo.
- ✓ Curva de aprendizaje favorable.
- ✓ Alto rendimiento.
- ✓ Amplia documentación.
- ✓ Gran comunidad de desarrollo.

Con las características planteadas se decide **React JS** como librería de interface de usuario. Para combinar con otras tecnologías necesarias para el desarrollo de la investigación, se asume la tecnología **MERN Stack** por agregar **React JS** en su entorno de trabajo.

### 1.5. Lenguaje de programación

✓ **JavaScript**

Es un lenguaje interpretado usado con 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 viejas limitaciones de rendimiento y confirmar el lenguaje JavaScript 1.8 como la mejor opción para la web. [11]

✓ **HTML**

HTML (lenguaje de marcas de hipertexto) es un lenguaje de marcado que establece una estructura para dar un sentido al contenido web, por ejemplo, define párrafos, cabeceras, tablas, imágenes y videos en la página. Para el desarrollo del módulo se utiliza HTML 4.01. [39]

Se selecciona *JavaScript* para el desarrollo de la aplicación por ser la base del entorno planteado por el autor y HTML por ser el lenguaje utilizado para el desarrollo de las vistas.

## 1.6. Herramientas a utilizar

✓ **NoSQLBooster 5.1.14**

Es un software para trabajar con *MongoDB* con una interfaz gráfica que ofrece una verdadera experiencia de sentido inteligente, que contiene completamiento de código. Tiene un servicio incorporado de lenguaje que infiere la mayoría de 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 comportamiento nativo de esquema para describir los tipos de datos, estructura de la colección y las validaciones. Con *Schema Analyzer Tool*, es posible tener un documento en que se puede definir esquemas de la colección y determinar tipos de datos de los atributos. [12]

✓ **Visual Studio Code 1.39.2**

Es un editor de código fuente ligero pero potente y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js. Cuenta con un amplio ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP, Go) y marcos de trabajos 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 soporte para sistema de control de versiones manteniendo excelente integración con Git. [13]

#### ✓ **Visual Paradigm 15.1**

Visual Paradigm es una herramienta para visualizar y diseñar elementos de software, para ello utiliza UML en su versión 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. [15]

Una vez conceptualizadas las principales herramientas se listan las que serán utilizadas para el desarrollo de la solución propuesta:

- ✓ **Entorno de desarrollo:** Visual Studio Code 1.39.2.
- ✓ **Herramienta de Modelado:** Visual Paradigm 15.1
- ✓ **Ciente para interacción con la base de datos:** NoSQLBooster 5.1.14.
- ✓ **Marco de trabajo:** MERN Stack.

## **1.7. 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. [23] Estas se clasifican en tradicionales y ágiles.

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, hacen especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso. [24]

RUP es un proceso formal: Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por *Rational Software*, y está integrado con toda la suite *Rational* de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte (personalización). Es guiado por casos de uso y centrado en la arquitectura, y utiliza UML como lenguaje de modelado. Realiza procesos de continuas pruebas y retroalimentación. Sin embargo, los beneficios obtenidos recompensan el esfuerzo invertido en este aspecto. El proceso de desarrollo constituye un marco metodológico que define en términos de metas estratégicas, objetivos, actividades y artefactos (documentación) requerido en cada fase de desarrollo. Esto permite enfocar esfuerzo de los recursos humanos en términos de habilidades, competencias y capacidades a asumir roles específicos con responsabilidades bien definidas. [24]

*Scrum* es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nuevas funcionalidades. Las iteraciones en general tienen una duración entre 2 y 4 semanas. [24]

Por su parte XP es lo más destacado de los procesos ágiles de desarrollo de software formulados por Kent Beck. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Esta metodología considera los cambios de requisitos sobre la marcha como un aspecto natural, inevitable e incluso deseable para el desarrollo de proyectos. Plantea que ser capaces de adaptarse a los cambios de requerimientos que puedan surgir sobre cualquier etapa de desarrollo constituye una aproximación mejor y más realista del producto que necesita el cliente para satisfacer sus necesidades. [24]

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

- ✓ Teniendo en cuenta las características del equipo de desarrollo, resulta apropiado la implementación de prácticas correspondientes a la metodología XP, la permanente interacción con el cliente, la refactorización constante, entre otras ventajas que brinda dicha metodología.

- ✓ La aplicación que se desarrolla es muy práctica y guarda mucha relación con los procesos que realiza el cliente diariamente, por lo que no requiere demasiada documentación sino buena calidad y fácil uso para su puesta en explotación.
- ✓ XP reduce el número de cambios necesarios para el proyecto lo que permite mejorar el tiempo de implementación. Por otra parte, existe propiedad de código compartido y simplicidad en el código lo que permite que este sea más entendible y modificable por los desarrolladores.

A raíz del análisis realizado y teniendo en cuenta que el cliente es tutor del proyecto, se puede afirmar que el sistema a desarrollar puede culminarse con resultados exitosos aplicando la metodología XP.

### **1.8. Conclusiones del capítulo**

- ✓ Los sistemas para la gestión estudiantil estudiados poseen funcionalidades como listar estudiantes, elaboración de reportes, sistema de autenticación para garantizar la disponibilidad y confidencialidad, pero se han identificado las siguientes limitaciones: sus funcionalidades no responden al negocio que se maneja en el SAE, no cumplen con requisitos de rendimiento planteados por el autor. Excepto el sistema informático implementado en el SAE, todos los sistemas estudiados son multiplataforma, característica que se tendrá en cuenta para el desarrollo de la presente investigación.
- ✓ Para la selección de las herramientas y tecnologías para el desarrollo de la aplicación, se tuvieron en cuenta principalmente el dominio por parte del autor y sus características aplicadas al marco de trabajo seleccionado.
- ✓ Una vez estudiado y analizado un conjunto de tecnologías actuales, el autor selecciona un marco de trabajo basado en el lenguaje JavaScript, entre las que se encuentra el marco de trabajo MERN Stack, Node Js y como cliente para interactuar con la base de datos NoSQLBooster.
- ✓ Como herramientas para el desarrollo se seleccionan: Visual Paradigm para la generación de artefactos, como editor de código fuente Visual Studio Code. El capítulo concluye argumentando la selección de la metodología de desarrollo XP.

## **CAPÍTULO 2: PROPUESTA DE SOLUCIÓN**

En este capítulo se presenta la propuesta de solución del Módulo Reporte Estadístico del Sistema de Gestión Estudiantil, con la finalidad de gestionar información y brindar oportunamente los reportes y estadísticas relacionadas a los estudiantes angoleños cursando estudios en Cuba. Se muestran artefactos propios de la metodología seleccionada XP, 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. Se hace referencia a los niveles de acceso que tendrán los usuarios del sistema, el modelo de datos, se muestra la lista de reserva del producto. Se realiza el modelo conceptual e iteraciones para la producción de entregables del sistema.

### **2.1. Descripción de la propuesta de Solución**

Para dar solución a la problemática antes planteada, como parte del sistema informático a desarrollar, es necesario la implementación del módulo Configuración donde se realizan las siguientes operaciones: Gestionar universidades, Carreras, Áreas de formación y Usuarios. Dichas funcionalidades son de vital importancia pues estas tributan información imprescindible al módulo de reportes estadísticos. Se implementará el módulo estadísticas que brinda la opción de imprimir datos cuantitativos de los estudiantes aplicando los filtros disponibles y la combinación de estos. El módulo reporte que permite listar y exportar los datos de los estudiantes. Es válido aclarar que todo lo referente a la gestión de estudiantes y su información docente, se está desarrollando en un trabajo de culminación de estudios en la Universidad de las Ciencias Informáticas.

Los usuarios tendrán la facilidad de exportar los datos estadísticos en formato Excel, estos datos se podrán enviar al INAGBE para que se tenga la información actualizada de los estudiantes, ayudando así en la toma de decisiones para la ubicación laboral. A continuación, se muestra una ilustración de la propuesta de solución. La figura 3 es un prototipo no funcional que ilustra el listado de los estudiantes.

SGE Sistema de Gestão Estudantil Nome do usuário

Página principal  
Estudantes  
Actividades  
Estatísticas  
Associação  
Usuários  
Configurações

### Estudantes

Lista de estudantes

Cadastrar estudante

Pesquisar...

Nome	Ano académico	Universidade	Provincia residente	Curso	Opções
Eldade Joaquim Bondo	3º	UCI	Havana	Eng Informatica	Vizualizar

« < 1 > »

Copyright 2018 Estudantes Angolanos da UCI

Figura 4. Listado de los estudiantes

## 2.2. Usuarios del Sistema

Como parte de la seguridad implementada para una mejor comprensión los usuarios del sistema son clasificados por niveles. A continuación, se explica brevemente el alcance de acceso dentro del sistema de estos niveles.

**Nivel 0:** Los usuarios de nivel 0 son los que tienen acceso total al sistema, tienen acceso a todas las funcionalidades. En tal caso puede ser el director del SAE o un funcionario que este designe.

**Nivel 1:** Los usuarios de nivel 1 no tienen acceso a todas las funcionalidades que el sistema brinda, solo las definidas previamente por el administrador y a su perfil de usuario donde podrá modificar determinada información propia.

### 2.3. Lista de reserva del producto

Con el objetivo de realizar las tareas de forma organizada, en un proyecto de software se realiza la lista de reserva del producto. Siendo un artefacto típico de la metodología XP que constituye una opción para dar prioridad a las funcionalidades a implementar. A continuación, se muestra la lista de reserva de producto confeccionada para el desarrollo de las funcionalidades presentes en la propuesta de solución.

Tabla 4. Lista de reserva del producto

Ítem	Descripción
<b>Prioridad: Alta</b>	
1	Generar datos estadísticos de pregrado por carrera y género
2	Exportar datos estadísticos de pregrado por carrera y género
3	Generar datos estadísticos de pregrado por carrera y año
4	Exportar datos estadísticos de pregrado por carrera y año
5	Generar proyección de graduación
6	Exportar proyección de graduación
7	Generar datos estadísticos de pregrado por provincia de origen
8	Exportar datos estadísticos de pregrado por provincia de origen
9	Generar datos estadísticos de pregrado por provincia de Cuba
10	Exportar datos estadísticos de pregrado por provincia de Cuba
11	Generar datos estadísticos de pregrado por provincia y carrera
12	Exportar datos estadísticos de pregrado por provincia y carrera
13	Listar estudiantes de pregrado
14	Exportar estudiantes de pregrado
15	Filtrar estudiantes de pregrado
16	Listar estudiantes de postgrado
17	Exportar estudiantes de postgrado
18	Filtrar estudiantes de postgrado
19	Listar los estudiantes finalistas

20	Exportar los estudiantes finalistas
21	Filtrar los estudiantes finalistas
22	Listar estudiantes que causaron baja académica
23	Exportar estudiantes que causaron baja académica
24	Filtrar estudiantes que causaron baja académica
<b>Prioridad: Media</b>	
25	Generar datos estadísticos de postgrado por provincia
26	Exportar datos estadísticos de postgrado por provincia
27	Generar datos estadísticos de postgrado por año
28	Exportar datos estadísticos de postgrado por año
29	Generar datos estadísticos de finalistas por carrera y género
30	Exportar datos estadísticos de finalistas por carrera y género
31	Generar datos estadísticos de finalistas por provincia
32	Exportar datos estadísticos de finalistas por provincia
33	Generar datos estadísticos de finalistas por provincia y carrera
34	Exportar datos estadísticos de finalistas por provincia y carrera
<b>Prioridad: Baja</b>	
35	Generar gráfico de proyección de graduación
36	Generar gráfico de estudiantes por provincia
37	Generar gráfico de estudiantes de pregrado por año
38	Generar gráfico de estudiantes de pregrado por área de formación

## 2.4. Historias de Usuario (HU)

Las HU describen las funcionalidades que deben ser implementadas en el sistema, por tal razón se dice que es el artefacto eje de la metodología XP. Las mismas son escritas por el cliente usando un lenguaje transparente y conciso que deje bien claro las necesidades a suplir con el sistema. Permiten darle respuesta inmediatamente a las funcionalidades cambiantes en el transcurso del desarrollo y sirven de base para las pruebas funcionales del sistema. Se emplean para estimar el tiempo y para la realización de los planes de entregas. En aras de concretar lo planificado, se decide dividir por iteraciones:

Iteración 1: Se trabaja en los requisitos de estadísticas por tablas para los estudiantes de pregrado.

Iteración 2: En esta iteración se realizan las HU relacionadas a los estudiantes de postgrado y los finalistas.

Iteración 3: Se realizan las HU relacionadas con la generación de estadísticas por gráfico.

Durante la fase de exploración se realizaron HU que abarcan las funcionalidades correspondientes al sistema propuesto, a continuación, se muestran algunas de ellas.

*Tabla 5. HU-1 Generar datos estadísticos de pregrado por carrera y género*

<b>Historia de Usuario 1</b>	
Nombre: Generar datos estadísticos de pregrado por carrera y género.	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alto
Puntos Estimados: 5	Puntos Reales: 4
Descripción: Permite crear una tabla donde es posible saber la cantidad de estudiantes por carrera y género.	
Observación: Todos los usuarios tienen acceso a dicha funcionalidad.	

*Tabla 6. HU-13 Listar estudiantes de pregrado*

<b>Historia de Usuario 13</b>	
Nombre: Listar estudiantes de pregrado	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alto
Puntos Estimados: 5	Puntos Reales: 6
Descripción: Permite listar todos los estudiantes de pregrado en una tabla que sea posible aplicar filtros.	
Observación: Todos los usuarios tienen acceso a dicha funcionalidad.	

*Tabla 7. HU-15 Filtrar estudiantes de pregrado*

<b>Historia de Usuario 15</b>	
Nombre: Filtrar estudiantes de pregrado	Iteración Asignada: 1

Prioridad en Negocio: Alta	Riesgos de Desarrollo: Alto
Puntos Estimados: 5	Puntos Reales: 6
Descripción: Permite al usuario realizar filtrados para facilitar las búsquedas, los filtros pueden ser por provincia, universidad, carrera u otros atributos.	
Observación: Todos los usuarios tienen acceso a dicha funcionalidad.	

*Tabla 8. HU-35 Generar gráfico de proyección de graduación general*

<b>Historia de Usuario 35</b>	
Nombre: Generar gráfico de proyección de graduación	Iteración Asignada: 3
Prioridad en Negocio: Baja	Riesgos de Desarrollo: Bajo
Puntos Estimados: 5	Puntos Reales: 4
Descripción: Usando gráficos como una mejor forma de ilustrar las estadísticas, se genere un gráfico mostrando la proyección de graduación en los próximos cinco años.	
Observación: Todos los usuarios tienen acceso a dicha funcionalidad.	

*Tabla 9. HU-38 Generar gráfico de estudiantes de finalistas por provincia*

<b>Historia de Usuario 38</b>	
Nombre: Generar gráfico de estudiantes finalistas por provincia	Iteración Asignada: 3
Prioridad en Negocio: baja	Riesgos de Desarrollo: medio
Puntos Estimados: 2	Puntos Reales: 3
Descripción: Usando elementos gráficos, se puede ver la cantidad de estudiantes finalistas distribuidos por provincias de Angola.	
Observación: Todos los usuarios tienen acceso a dicha funcionalidad.	

Tabla 10. HU-20 Exportar estudiantes finalistas

<b>Historia de Usuario 20</b>	
Nombre: Exportar estudiantes finalistas	Iteración Asignada: 3
Prioridad en Negocio: Alta	Riesgos de Desarrollo: medio
Puntos Estimados: 2	Puntos Reales: 3
Descripción: Permite exportar el listado de los estudiantes finalistas para un Excel.	
Observación: Todos los usuarios tienen acceso a dicha funcionalidad.	

## 2.5. Requisitos no funcionales

Los requisitos no funcionales son propiedades que hacen al producto atractivo, usable, rápido y confiable. Se refieren a todos los requisitos que no describen información a guardar o funciones a realizar. Se conocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el software. [26]

Tabla 11. Listado de requisitos no funcionales del sistema

<b>Usabilidad</b>	
<b>RNF1</b>	El sistema será utilizado por todo el personal que tenga acceso a la red del SAE y necesite hacer uso del mismo.
<b>RNF2</b>	El sistema debe garantizar un acceso fácil y podrá ser usado por usuarios con pocos conocimientos informáticos.
<b>RNF3</b>	Los usuarios deben de autenticarse para poder acceder a las a las funcionalidades del sistema.
<b>RNF4</b>	Mostrar elementos de apoyo a la navegación.
<b>Eficiencia</b>	
<b>RNF5</b>	El sistema debe responder en un tiempo menor a los 7 segundos
<b>Soporte</b>	
<b>RNF6</b>	El sistema debe recibir mantenimiento ante cualquier fallo que ocurra.
<b>RNF7</b>	El sistema debe ser escalable, para que en un futuro se incorpore nuevas funcionalidades en caso de ser necesarias.
<b>Hardware</b>	

<b>RNF8</b>	Debe existir una red de área local para permitir el acceso al sistema.
<b>RNF9</b>	<ul style="list-style-type: none"> <li>• El Servidor de Bases de Datos y Servidor Web deben tener: Microprocesador de 8 núcleos, 4GB de memoria RAM, 250 GB de disco duro como mínimo.</li> <li>• El cliente debe tener: Microprocesador de 1GHz, 1GB de memoria RAM, 512 GB de disco duro como mínimo.</li> </ul>
<b>Portabilidad</b>	
<b>RNF10</b>	El sistema debe ser multiplataforma
<b>Apariencia o interface externa</b>	
<b>RNF11</b>	Interface sencilla y amigable.
<b>RNF12</b>	La interfaz contará con menús para acceder a cada una de las vistas del sistema.

## 2.6. Planeación

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 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 HU es 3). [22]

Teniendo en cuenta que un punto de estimación equivale a una semana, las HU tienen como máximo 3 puntos para implementación. Para las HU se estimarán en días, donde los días deben estar entre 1 a 21 días para que no sea mayor que 3 puntos.

### 2.6.1. Estimación de Esfuerzo por Historias de Usuario

Para que se cumpla con lo planeado, se realizó una estimación por HU identificadas y definidas anteriormente para alcanzar una medida, lo más cercana a la realidad, del progreso del proyecto. Como consecuencia de este proceso se obtuvo el siguiente resultado:

*Tabla 12. Estimación de Esfuerzo por HU*

Ítem	Descripción	Duración(días)
1	Generar datos estadísticos de pregrado por carrera y género	2

<b>2</b>	Exportar datos estadísticos de pregrado por carrera y género	1
<b>3</b>	Generar datos estadísticos de pregrado por carrera y año	1
<b>4</b>	Exportar datos estadísticos de pregrado por carrera y año	1
<b>5</b>	Generar proyección de graduación	3
<b>6</b>	Exportar proyección de graduación	1
<b>7</b>	Generar datos estadísticos de pregrado por provincia de origen	2
<b>8</b>	Exportar datos estadísticos de pregrado por provincia de origen	1
<b>9</b>	Generar datos estadísticos de pregrado por provincia de Cuba	2
<b>10</b>	Exportar datos estadísticos de pregrado por provincia de Cuba	1
<b>11</b>	Generar datos estadísticos de pregrado por provincia y carrera	2
<b>12</b>	Exportar datos estadísticos de pregrado por provincia y carrera	1
<b>13</b>	Listar estudiantes de pregrado	1
<b>14</b>	Exportar estudiantes de pregrado	1
<b>15</b>	Filtrar estudiantes de pregrado	3
<b>16</b>	Listar estudiantes de postgrado	2
<b>17</b>	Exportar estudiantes de postgrado	1
<b>18</b>	Filtrar estudiantes de postgrado	1
<b>19</b>	Listar estudiantes finalistas	1
<b>20</b>	Exportar estudiantes finalistas	1
<b>21</b>	Filtrar estudiantes finalistas	1
<b>22</b>	Listar estudiantes con baja académica	1
<b>23</b>	Exportar estudiantes con baja académica	1
<b>24</b>	Filtrar estudiantes con baja académica	1
<b>25</b>	Generar datos estadísticos de postgrado por provincia	2
<b>26</b>	Exportar datos estadísticos de postgrado por provincia	1
<b>27</b>	Generar datos estadísticos de postgrado por año	2
<b>28</b>	Exportar datos estadísticos de postgrado por año	1
<b>29</b>	Generar datos estadísticos de finalistas por carrera y género	2
<b>30</b>	Exportar datos estadísticos de finalistas por carrera y género	1

<b>31</b>	Generar datos estadísticos de finalistas por provincia	2
<b>32</b>	Exportar datos estadísticos de finalistas por provincia	1
<b>33</b>	Generar datos estadísticos de finalistas por provincia y carrera	2
<b>34</b>	Exportar datos estadísticos de finalistas por provincia y carrera	1
<b>35</b>	Generar gráfico de proyección de graduación	3
<b>36</b>	Generar gráfico de estudiantes por provincia	3
<b>37</b>	Generar gráfico de estudiantes de pregrado por año	3
<b>38</b>	Generar gráfico de estudiantes de pregrado por área de formación	3

### 2.6.2. Plan e 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. [23] 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. [22]

El plan de entregas se basa en las Historias de Usuario. Una vez identificadas las mismas, se definen los niveles de prioridad lo que permitió dividir el desarrollo en 3 iteraciones que fueron mencionadas en el epígrafe 2.4 de acuerdo con la prioridad de cada HU y la relación en cuanto a funcionalidad entre ellas.

*Tabla 13. Estimación de esfuerzo por HU calculados*

<b>Iteración</b>	<b>Descripción</b>	<b>Duración(días)</b>	<b>Semanas</b>
<b>1</b>	Generar datos estadísticos de pregrado por carrera y género	2	4
	Exportar datos estadísticos de pregrado por carrera y género	1	

Generar datos estadísticos de pregrado por carrera y año	1	
Exportar datos estadísticos de pregrado por carrera y año	1	
Generar proyección de graduación	3	
Exportar proyección de graduación	1	
Generar datos estadísticos de pregrado por provincia de origen	2	
Exportar datos estadísticos de pregrado por provincia de origen	1	
Generar datos estadísticos de pregrado por provincia de Cuba	2	
Exportar datos estadísticos de pregrado por provincia de Cuba	1	
Generar datos estadísticos de pregrado por provincia y carrera	2	
Exportar datos estadísticos de pregrado por provincia y carrera	1	
Listar estudiantes de pregrado	1	
Exportar estudiantes de pregrado	1	
Filtrar estudiantes de pregrado	3	
Listar estudiantes de postgrado	2	
Exportar estudiantes de postgrado	1	
Filtrar estudiantes de postgrado	1	
Listar estudiantes finalistas	1	
Exportar estudiantes finalistas	1	
Filtrar estudiantes finalistas	1	
Listar estudiantes con baja académica	1	
Exportar estudiantes con baja académica	1	

	Filtrar estudiantes con baja académica	1	
<b>2</b>	Generar datos estadísticos de postgrado por provincia	2	2
	Exportar datos estadísticos de postgrado por provincia	1	
	Generar datos estadísticos de postgrado por año	2	
	Exportar datos estadísticos de postgrado por año	1	
	Generar datos estadísticos de finalistas por carrera y género	2	
	Exportar datos estadísticos de finalistas por carrera y género	1	
	Generar datos estadísticos de finalistas por provincia	2	
	Exportar datos estadísticos de finalistas por provincia	1	
	Generar datos estadísticos de finalistas por provincia y carrera	2	
	Exportar datos estadísticos de finalistas por provincia y carrera	1	
<b>3</b>	Generar gráfico de proyección de graduación	3	2
	Generar gráfico de estudiantes por provincia	3	
	Generar gráfico de estudiantes de pregrado por año	3	
	Generar gráfico de estudiantes de pregrado por área de formación	3	

<b>Total</b>	59 días ≈ 2 meses
--------------	-------------------

## 2.7. Diseño de la Aplicación

### 2.7.1. 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. [25] 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 las que se exponen a continuación:

*Tabla 14. Tarjeta CRC Estudiante*

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

Tabla 15. Tarjeta CRC Graduado

<b>Tarjeta CRC</b>	
<b>Clase:</b> Graduado	
<b>Responsabilidad</b>	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>	Estudiante

Tabla 16. Tarjeta CRC Baja

<b>Tarjeta CRC</b>	
<b>Clase:</b> Baja	
<b>Responsabilidad</b>	Colaboración
<p>Contiene toda la información de un estudiante dado por baja.</p> <p>Permite adicionar un estudiante dado por baja, cambiar los atributos de un estudiante.</p> <p>Permite filtrar o buscar</p>	<p>Estudiante</p> <p>Motivo de Baja</p>

Tabla 17. Tarjeta CRC Resultado académico

<b>Tarjeta CRC</b>	
<b>Clase:</b> Resultado académico	
<b>Responsabilidad</b>	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 18. Tarjeta CRC Carrera

<b>Tarjeta CRC</b>	
<b>Clase:</b> Carrera	
<b>Responsabilidad</b>	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>	Área de formación

Tabla 19. Tarjeta CRC Universidad

<b>Tarjeta CRC</b>	
<b>Clase:</b> Universidad	
<b>Responsabilidad</b>	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>	Provincia

Tabla 20. Tarjeta CRC Provincia

<b>Tarjeta CRC</b>	
<b>Clase:</b> Provincia	
<b>Responsabilidad</b>	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>Permite listar y eliminar</p>	País

Tabla 21. Tarjeta CRC Área de Formación

Tarjeta CRC	
<b>Clase:</b> Área de Formación	
<b>Responsabilidad</b>	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 22. Tarjeta CRC País

Tarjeta CRC	
<b>Clase:</b> País	
<b>Responsabilidad</b>	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 23. Tarjeta CRC Motivo de Baja

Tarjeta CRC	
<b>Clase:</b> Motivo de Baja	
<b>Responsabilidad</b>	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 24. Tarjeta CRC Usuario

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

## 2.8. Arquitectura

Se decide determinar el patrón arquitectónico Modelo-Vista-Controlador. Este patrón separa la lógica de negocio en tres capas 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, por lo general 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 de la aplicación 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. [15]

La figura 4 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*.

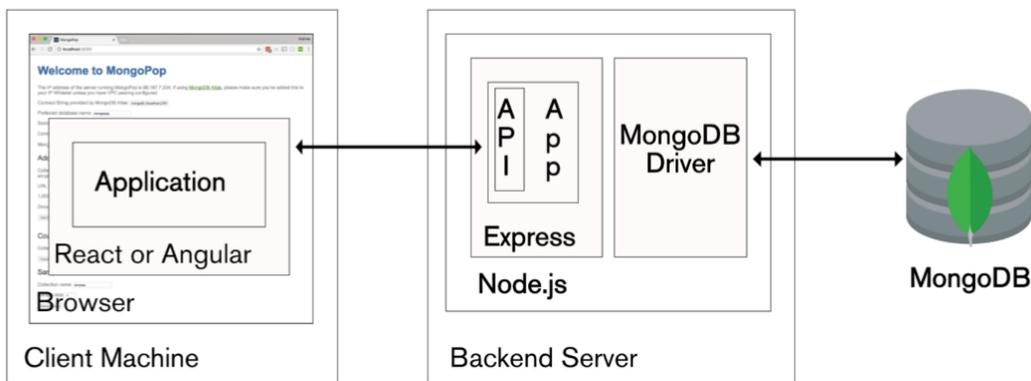


Figura 5. Diseño conceptual de la arquitectura

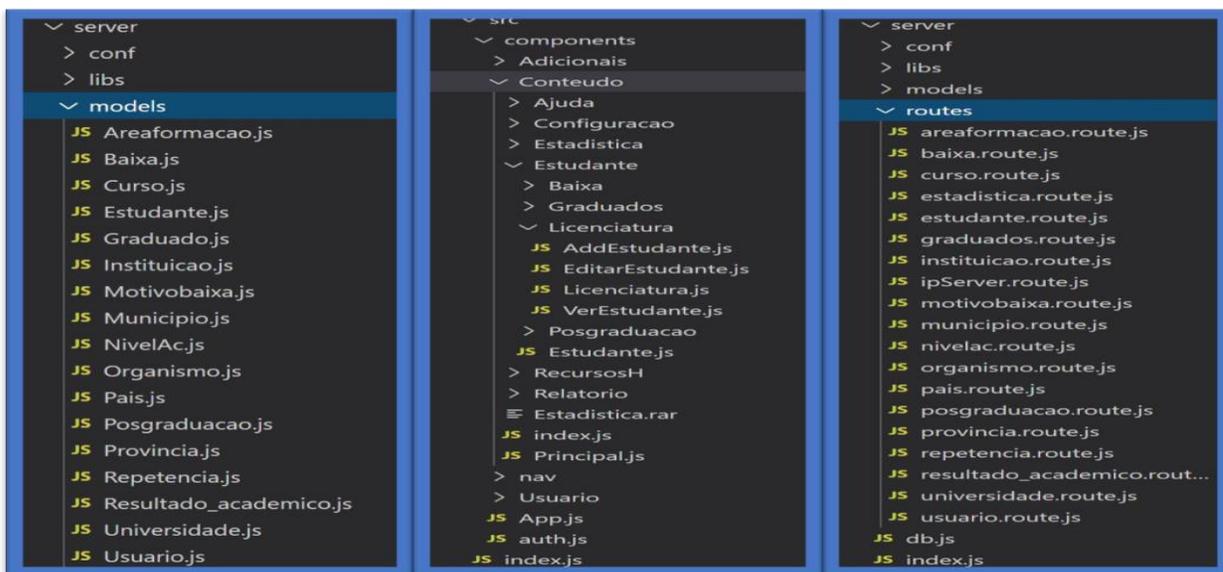


Figura 6. Evidencia de la arquitectura

## 2.9. 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. [27] 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.

### Patrones GoF (*Gang of Four*)

En el año 1994, se publica el libro “*Design Patterns: Elements of Reusable Object Oriented Software*” escrito por *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. [28]

Para el desarrollo de aplicaciones web, se identificaron 8 patrones de diseño Gof de los 23 en los procesos de desarrollo *software*. [29] A continuación se muestran los que se utilizaron en el presente trabajo:

#### Creacionales:

Constructora (*Builder*): un patrón de creación de objetos delega la instanciación en otro objeto. [30]

```
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
```

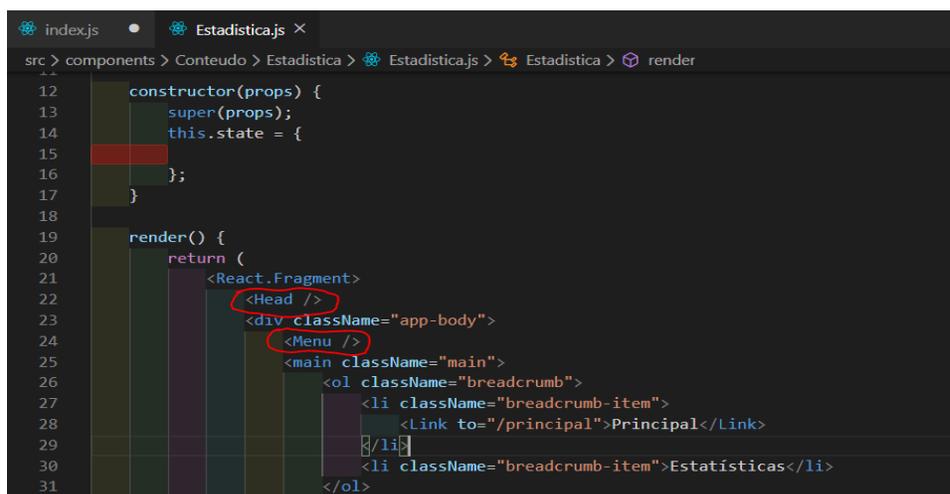
Figura 7. Evidencia de la arquitectura.

## Estructurales:

Decorador (Decorator): Añade funcionalidad a una clase dinámicamente. [29]

Adaptador (Adapter): utiliza la herencia para componer interfaces o implementaciones. [30]

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 Submenu.js que son usados en diversas partes del sistema.



```
12 constructor(props) {
13   super(props);
14   this.state = {
15   };
16 };
17 }
18
19 render() {
20   return (
21     <React.Fragment>
22       <Head />
23       <div className="app-body">
24         <Menu />
25         <main className="main">
26           <ol className="breadcrumb">
27             <li className="breadcrumb-item">
28               <Link to="/principal">Principal</Link>
29             </li>
30             <li className="breadcrumb-item">Estatísticas</li>
31           </ol>

```

Figura 8. Ejemplo del uso de herencia

## 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.

```

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 }
348 console.log(ident)
349 BuscarUniversidadeProv(ident, 'universidade')
350     .then(data => {
351         this.setState({ universidadeApi: data })
352         console.log(data)
353     })
354 }
355
356

```

*Figura 9. Evidencia de la arquitectura.*

## 2.10- Conclusiones del capítulo

En este capítulo se hizo la propuesta de solución de la problemática con la implementación de un módulo para el sistema de gestión estudiantil. Se describieron los principales artefactos que propone la metodología XP. Se definieron las funcionalidades a implementar y los requisitos que debe cumplir el producto en las HU, así como la descripción de las historias de usuario divididas por tres iteraciones, la estimación de esfuerzo dedicado a la realización de cada una de ellas en el plan de entregas y las tarjetas CRC. Finaliza el capítulo con la descripción de la arquitectura a utilizar en el desarrollo. Se hace una breve referencia conceptual de los principales patrones de diseños empleados y se muestra su uso en fragmentos del código implementado en la arquitectura base.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

En el presente capítulo se analizan los aspectos relacionados con la implementación y las pruebas funcionales 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. Modelo de Datos

Un modelo es un conjunto de herramientas conceptuales para describir datos, sus relaciones, su significado y sus restricciones de consistencia. [31]

Antes de partir para el proceso de implementación, hubo la necesidad de realizar la conversión de la base de datos de *Access* para el formato *JSON* utilizando el programa *Ms Access to Json* en su versión 1.4. Posteriormente se procedió al proceso de limpieza de los datos utilizando *Pentaho Data Integration*. Vale destacar que este proceso solo se ha realizado con la entidad “*Estudiante*”, las demás entidades fueron creadas una por una siguiendo con las normas de creación de las bases de datos *noSQL*.

*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. [32]

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. [33] A continuación se muestra un fragmento de la estructura de los documentos.

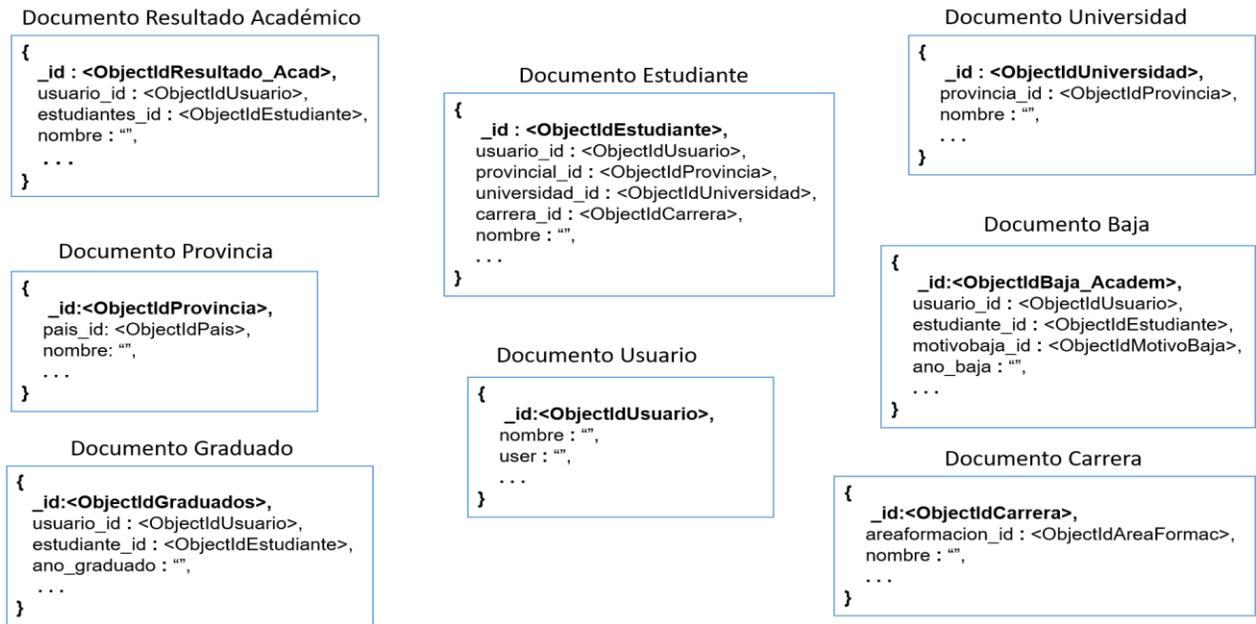


Figura 10. Evidencia del modelo.

*Mongodb* es una base de datos *noSQL* donde los documentos no tienen relación entre ellos, pero el contenido de los documentos se puede relacionar con los contenidos de otros documentos usando el *id*. [33]. Es evidente la relación en el documento estudiante con algunas entidades pues recibe el identificador de otros documentos. Abajo se ilustra una relación en *Mongodb*.

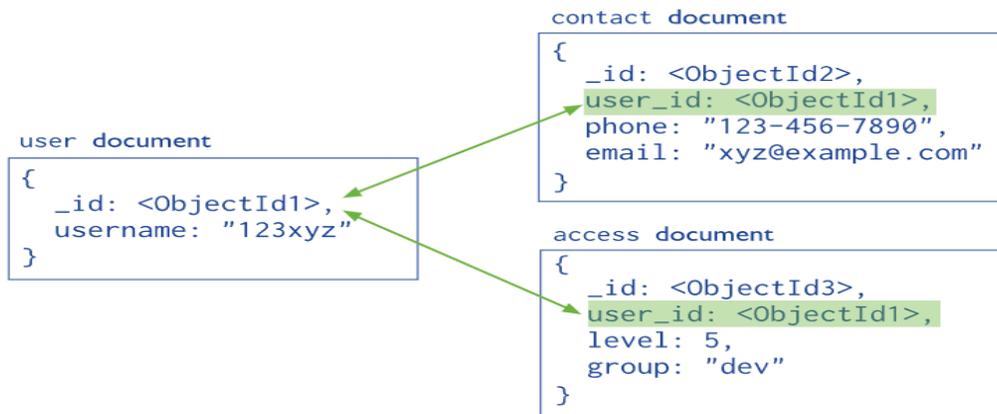


Figura 11. Ejemplo de relación entre documentos en *Mongodb*.

### 3.2. Tareas de Ingeniería

En la fase de planeación se definieron los elementos funcionales necesarios para realizar el desarrollo de la aplicación, lo que implica que se puede proceder a la fase de implementación, donde tareas de ingeniería es el artefacto usado. [36] Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

Para la **primera iteración** se definieron las siguientes tareas de ingeniería:

HU 5: Generar proyección de graduación.

✓ TI 1: Generar proyección de graduación.

HU 13: Listar estudiantes de pregrado.

✓ TI 2: Acceder los datos del API.

✓ TI 3: Listar estudiantes de pregrado activos.

HU 15: Filtrar estudiantes de pregrado.

✓ TI 4: Crear filtros usando caja de texto.

✓ TI 5: Crear filtros usando selectores.

HU 11: Generar datos estadísticos de pregrado por provincia y carrera.

✓ TI 6: Generar datos estadísticos de pregrado por provincia y carrera.

✓ TI 7: Listar los datos en una tabla.

HU 14: Exportar estudiantes de pregrado.

✓ TI 8: Exportar estudiantes de pregrado.

*Tabla 25. Generar proyección de graduación.*

<b>Tarea de ingeniería</b>	
<b>No. tarea:</b> 1	<b>Numero de HU:</b> 5
<b>Nombre de la tarea:</b> Generar proyección de graduación.	

<b>Tipo de tarea:</b> Desarrollo	Punto estimado: 5
<b>Fecha inicio:</b> 01/12/2019	Fecha fin: 03/12/2019
<b>Programador responsable:</b> Eldade Joaquim Bondo	
<b>Descripción:</b> Muestra la proyección de graduación para los próximos cinco años.	

*Tabla 26. Listar estudiantes de pregrado activos.*

<b>Tarea de ingeniería</b>	
<b>No. tarea:</b> 3	Numero de HU: 13
<b>Nombre de la tarea:</b> Listar estudiantes de pregrado activos.	
<b>Tipo de tarea:</b> Desarrollo	Punto estimado: 5
<b>Fecha inicio:</b> 05/12/2019	Fecha fin: 06/12/2019
<b>Programador responsable:</b> Eldade Joaquim Bondo	
<b>Descripción:</b> Muestra la lista de los estudiantes de pregrado activos.	

*Tabla 27. Filtrar estudiantes de pregrado*

<b>Tarea de ingeniería</b>	
<b>No. tarea:</b> 5	Numero de HU: 15
<b>Nombre de la tarea:</b> Filtrar estudiantes de pregrado.	
<b>Tipo de tarea:</b> Desarrollo	Punto estimado: 5
<b>Fecha inicio:</b> 07/12/2019	Fecha fin: 08/12/2019
<b>Programador responsable:</b> Eldade Joaquim Bondo	
<b>Descripción:</b> Permite al usuario realizar filtrados para facilitar las búsquedas, los filtros pueden ser por provincia, universidad, carrera u otros atributos.	

Para la **segunda iteración** se definieron las siguientes tareas de ingeniería:

HU 33: Generar datos estadísticos de finalistas por provincia y carrera.

- ✓ TI 9: Crear una tabla con datos estadísticos de estudiantes finalistas por provincia y carrera.

HU 34: Exportar datos estadísticos de finalistas por provincia y carrera.

- ✓ TI 10: Exportar para Excel la tabla con datos estadísticos de estudiantes finalistas por provincia y carrera.

HU 25: Generar datos estadísticos de postgrado por provincia.

- ✓ TI 11: Crear una tabla con datos estadísticos de estudiantes de postgrado por provincia.

HU 26: Exportar datos estadísticos de postgrado por provincia.

- ✓ TI 11: Exportar para Excel la tabla con datos estadísticos de estudiantes de postgrado por provincia.

*Tabla 28. Estadística de estudiantes finalistas*

<b>Tarea de ingeniería</b>	
<b>No. tarea:</b> 9	<b>Numero de HU:</b> 33
<b>Nombre de la tarea:</b> Crear una tabla con datos estadísticos de estudiantes finalistas por provincia y carrera.	
<b>Tipo de tarea:</b> Desarrollo	<b>Punto estimado:</b> 5
<b>Fecha inicio:</b> 10/12/2019	<b>Fecha fin:</b> 11/12/2019
<b>Programador responsable:</b> Eldade Joaquim Bondo	
<b>Descripción:</b> Listar datos cuantitativos de los estudiantes finalistas por provincia y carrera.	

*Tabla 29. Exportar tabla de estudiantes finalistas por provincia y carrera*

<b>Tarea de ingeniería</b>	
<b>No. tarea:</b> 10	<b>Numero de HU:</b> 34
<b>Nombre de la tarea:</b> Exportar para Excel la tabla con datos estadísticos de estudiantes finalistas por provincia y carrera.	
<b>Tipo de tarea:</b> Desarrollo	<b>Punto estimado:</b> 5
<b>Fecha inicio:</b> 15/12/2019	<b>Fecha fin:</b> 16/12/2019
<b>Programador responsable:</b> Eldade Joaquim Bondo	

**Descripción:** Exportar para Excel la tabla de datos.

*Tabla 30. Estadísticas de los estudiantes de postgrado por provincia.*

<b>Tarea de ingeniería</b>	
<b>No. tarea:</b> 11	Numero de HU: 25
<b>Nombre de la tarea:</b> Crear una tabla con datos estadísticos de estudiantes de postgrado por provincia.	
<b>Tipo de tarea:</b> Desarrollo	Punto estimado: 4
<b>Fecha inicio:</b> 19/12/2019	Fecha fin: 20/12/2019
<b>Programador responsable:</b> Eldade Joaquim Bondo	
<b>Descripción:</b> Listar datos cuantitativos de los estudiantes de postgrado por provincia.	

Para la **tercera iteración** se definieron las siguientes tareas de ingeniería:

HU 35: Generar gráfico de proyección de graduación.

- ✓ TI 12: Generar gráfico de proyección de graduación para estudiantes de medicina.
- ✓ TI 13: Generar gráfico de proyección de graduación para estudiantes de otras carreras.
- ✓ TI 13: Generar gráfico de proyección de graduación general.

HU 36: Generar gráfico de estudiantes por provincia.

- ✓ TI 14: Generar gráfico de estudiantes de pregrado por provincia.
- ✓ TI 15: Generar gráfico de estudiantes de postgrado por provincia.
- ✓ TI 16: Generar gráfico de estudiantes finalistas por provincia.

HU 37: Generar gráfico de estudiantes de pregrado por año.

- ✓ TI 17: Generar gráficos con la distribución cuantitativa de estudiantes por año.

HU 38: Generar gráfico de estudiantes de pregrado por área de formación.

- ✓ TI 18: Generar gráficos con la distribución cuantitativa de estudiantes por área de formación.

Tabla 31. Gráfico de los estudiantes de pregrado por año.

<b>Tarea de ingeniería</b>	
<b>No. tarea: 17</b>	Numero de HU: 37
<b>Nombre de la tarea: Generar gráfico de estudiantes de pregrado por año</b>	
<b>Tipo de tarea: Desarrollo</b>	Punto estimado: 4
<b>Fecha inicio: 24/12/2019</b>	Fecha fin: 25/12/2019
<b>Programador responsable: Eldade Joaquim Bondo</b>	
<b>Descripción: Muestra un gráfico con la distribución de los estudiantes por año.</b>	

### 3.3. Pruebas

Hoy en día, 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. [37]

El proceso de pruebas de software tiene dos objetivos fundamentales:

1. Demostrar al desarrollador y al cliente que el software cumple con los requisitos.
2. Descubrir defectos en el software en que el comportamiento de este, es incorrecto, no deseable o no cumple su especificación. [26]

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

#### 3.3.1 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. [38]

Las pruebas de caja blanca, denominadas a veces pruebas de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba

[37]. Con la aplicación de este método se garantiza que se verifique por lo menos una vez todos los caminos independientes de cada módulo, que se ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsas, que se ejecuten todos los bucles en sus límites y con sus límites operacionales y se entrenen las estructuras internas de datos para asegurar su validez.

Se hace uso de la técnica del camino básico que permite derivar casos de prueba a partir de un conjunto de caminos independientes por los cuales puede circular el flujo de control.

### **Pasos para el desarrollo de la técnica del Camino básico:**

1. Se escoge el diseño o el código de la aplicación como base, y se dibuja el correspondiente grafo de flujo. Se obtiene la complejidad ciclomática del grafo de flujo.
2. Se determina la complejidad ciclomática del grafo resultante, posteriormente se determinan los casos de pruebas que permitan la ejecución de los caminos anteriores.
3. Se establece un conjunto básico de caminos linealmente independientes.
4. Se elaboran los casos de pruebas que permitirán la ejecución de cada camino del conjunto básico.

### **Notación del grafo de flujo:**

- **Nodo del grafo de flujo(N):** Es un círculo que representa las sentencias procedimentales.
- **Aristas(A):** Son las flechas o enlaces que representan el flujo de control del grafo.
- **Regiones:** Son las áreas delimitadas por las aristas y nodos. Al realizar el conteo de las regiones también se incluye el área exterior del grafo.
- **Complejidad ciclomática:** Se calcula a partir del grafo de control del software bajo análisis y mide, a grandes rasgos, el número de caminos independientes que pueden ocurrir en la ejecución, entre el inicio y el fin del software.

Se define como:  $V(G) = A - N + 2$ .

A continuación, se muestra un fragmento de código perteneciente al algoritmo desarrollado:

```

455 export const BuscarPorNombre = (est, nome) => {
456     var j = 0;
457     estudanteF = [];
458
459     for (let i = 0; i < est.length; i++) {
460         if (est[i].name !== null && est[i].name !== "") {
461             Nomecolecacao = SubstituirAcento(est[i].name.toLowerCase());
462             Nomeenviado = SubstituirAcento(nome.toLowerCase());
463
464             if (Nomecolecacao.indexOf(Nomeenviado) !== -1 && Nomecolecacao.indexOf(Nomeenviado) !== '') {
465                 estudanteF[j++] = est[i]
466             }
467         }
468     }
469     return estudanteF;
470 }

```

Figura 12. Fragmento de código de búsqueda por nombre.

Posteriormente se procede a la elaboración del grafo de flujo teniendo en cuenta dicha enumeración.

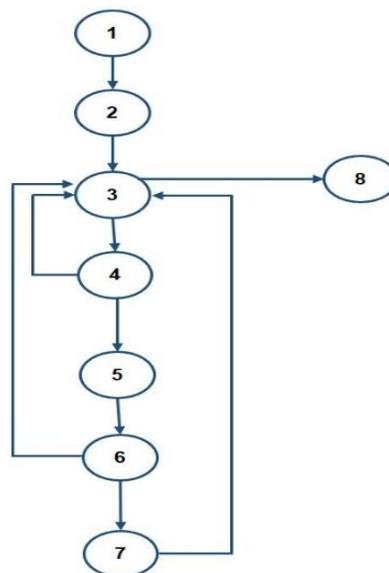


Figura 13. Representación del grafo de flujo de camino básico.

## Complejidad ciclomática

$$V(G) = A - N + 2$$

$$V(G) = 10 - 8 + 2 = 4$$

## Caminos independientes

1, 2, 3, 8

1, 2, 3, 4, 3, 8

1, 2, 3, 4, 5, 6, 3, 8

1, 2, 3, 4, 5, 6, 7, 3, 8

Cada camino independiente es un caso de prueba a realizar, de forma que se garantiza que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. En el caso anterior se calcularon 4 caminos básicos, por tanto, surge la necesidad de hacer igual número de casos de prueba. A continuación, se muestra los 4 casos de pruebas realizados.

*Tabla 32. Los casos de prueba realizados para el camino básico.*

<b>Caso de prueba 1 para el camino básico</b>	
<b>Descripción</b> Mostrar la tabla sin datos si la lista de estudiantes es vacía.	
<b>Condición de ejecución</b> Base de datos vacía.	
<b>Datos de entrada</b>	Nombre y apellidos.
<b>Tipo de datos esperado</b>	
<b>Evaluación del caso de Prueba : Satisfactoria</b>	
<b>Caso de prueba 2 para el camino básico</b>	
<b>Descripción</b> Mostrar la tabla sin datos si el campo nombre de la entidad estudiantes es vacío o nulo.	

<b>Condición de ejecución</b>	
Atributo nombre es nulo o vacío.	
<b>Datos de entrada</b>	Nombre y apellidos.
<b>Tipo de datos esperado</b>	
<b>Evaluación del caso de Prueba : Satisfactoria</b>	
Caso de prueba 3 para el camino básico	
<b>Descripción</b>	
Mostrar la tabla sin datos si no hay coincidencia de nombres.	
<b>Condición de ejecución</b>	
Nombres inexistentes	
<b>Datos de entrada</b>	Nombre y apellidos.
<b>Tipo de datos esperado</b>	
<b>Evaluación del caso de Prueba : Satisfactoria</b>	
Caso de prueba 4 para el camino básico	
<b>Descripción</b>	
Mostrar la tabla con datos de los estudiantes	
<b>Condición de ejecución</b>	
Nombres coincidentes	
<b>Datos de entrada</b>	Nombre y apellidos.
<b>Tipo de datos esperado</b>	
<b>Evaluación del caso de Prueba : Satisfactoria</b>	

### 3.3.2 Pruebas de Integración

Las pruebas de integración son una técnica sistemática para construir la arquitectura del software, al mismo tiempo, realizar pruebas para descubrir errores asociados con la interconexión. El objetivo es tomar componentes probados por unidades y construir la estructura del programa que se diseñó.

A menudo hay una tendencia a intentar la integración no incremental; es decir, para construir el programa usando un enfoque de "Big Bang". Todos los componentes se combinan de antemano. Todo el programa se prueba como un todo. Y el caos suele resultar. Se encuentra un conjunto de errores. La corrección es

difícil porque el aislamiento de las causas se complica por la vasta extensión de todo el programa. Una vez que se corrigen estos errores, aparecen otros nuevos y el proceso continúa en un bucle aparentemente interminable.

La integración incremental es la antítesis del enfoque del “Big Bang”. El programa se construye y se prueba en pequeños incrementos, donde los errores son más fáciles de aislar y corregir. Es más probable que las interfaces se prueben completamente y se puede aplicar un enfoque de prueba sistemático. [40]

Se emplea el método de integración incremental por las facilidades que brinda al desarrollador de realizar pruebas al programa menos complejas y obteniendo mejores resultados. A continuación, se presentan dos estrategias del método seleccionado.

### **Estrategias de integración**

**Top-down (descendente):** La prueba de integración descendente es un enfoque incremental para la construcción de la arquitectura del software. Los módulos se integran moviéndose hacia abajo a través de la jerarquía de control, comenzando con el módulo de control principal (programa principal). Los módulos subordinados al módulo de control principal se incorporan a la estructura ya sea en profundidad o primero en amplitud. El proceso de integración se realiza en una serie de cinco pasos:

1. El módulo de control principal se utiliza como controlador de prueba y los apéndices se sustituyen por todos los componentes directamente subordinados al módulo de control principal.
2. Dependiendo del enfoque de integración seleccionado (es decir, primero en profundidad o primero en amplitud), los apéndices subordinados se reemplazan uno a la vez con componentes reales.
3. Las pruebas se realizan a medida que se integra cada componente.
4. Al completar cada conjunto de pruebas, se reemplaza otro apéndice con el componente real.
5. Se pueden realizar pruebas de regresión para garantizar que no se hayan introducido nuevos errores.

El proceso continúa desde el paso 2 hasta que se construye toda la estructura del programa.

**Bottom-up (ascendente):** Las pruebas de integración ascendente, como su nombre lo indica, comienzan la construcción y las pruebas con módulos atómicos (es decir, componentes en los niveles más bajos en la

estructura del programa). Debido a que los componentes se integran de abajo hacia arriba, la funcionalidad proporcionada por los componentes subordinados a un nivel dado siempre está disponible y se elimina la necesidad de apéndices. Una estrategia de integración de abajo hacia arriba se puede implementar con los siguientes pasos:

1. Los componentes de bajo nivel se combinan en grupos (a veces llamados construcciones) que realizan una sub-función de software específica.
2. Un controlador (un programa de control para pruebas) está escrito para coordinar el ingreso y salida de prueba.
3. Se prueba el conjunto o construcción.
4. Los controladores se eliminan y los grupos se combinan moviéndose hacia arriba en la estructura del programa.

En este caso se emplea un acercamiento práctico (a veces llamado Sandwich testing) que utiliza Topdown para los niveles superiores de la estructura del programa y Bottom-up para los niveles inferiores. Combinando lo mejor de ambas técnicas. [40]

### **Detalles de la prueba**

La prueba se encuentra dividida en tres fases.

1. **Interacción de usuario:** En esta fase se revisan los datos de entrada y salida, el formato de presentación de los datos, el procesamiento de errores y la presentación de los mismos.
2. **Procesamiento de datos:** En esta fase se revisan la obtención de datos, manipulación del formato de los datos, determinación de condiciones de procesamiento, acciones requeridas como consecuencia de las condiciones encontradas.
3. **Comunicación:** En esta fase se revisan el traspaso de datos entre funciones y clases, la llamada a funciones, las rutas de comunicación entre clases y funciones.

En cada prueba se aplicaron los siguientes criterios:

- **Integridad de interfaz:** Son comprobadas las interfaces internas y externas de cada subsistema o funcionalidad que es integrada al módulo.

- **Validación funcional:** Se aplican pruebas diseñadas para descubrir errores funcionales.
- **Contenido de información:** Se aplican pruebas diseñadas para descubrir errores asociados con estructuras de datos locales o globales.
- **Desempeño:** Se aplican pruebas durante el diseño del software para verificar el desempeño del programa se encuentre entre los límites establecidos.

### Ejemplo de prueba de integración

Se aplica la prueba de integración al componente “Reporte de los estudiantes de pregrado”, obteniéndose los resultados que a continuación se muestran en cada fase.

*Tabla 33. Caso de prueba de integración del componente "Reporte de los estudiantes de pregrado".*

<b>Caso de prueba de integración del componente: Reporte de los estudiantes de pregrado</b>	
<b>Sistema al que se integra</b>	Sistema de gestión estudiantil
<b>Condiciones de ejecución</b>	Existe al menos un estudiante en la base de datos.
<b>Descripción de la prueba</b>	Comprobar que el componente “Reporte de los estudiantes de pregrado” proporcione la información referente a los estudiantes, que los filtros funciones correctamente, además que sea posible exportar el listado de los estudiantes a Excel.
<b>Pasos de ejecución</b>	El probador interactúa con el componente. Se selecciona el criterio de búsqueda. Se realiza la exportación al Excel. Comprobar en el fichero Excel que los datos ha sido exportados correctamente.
<b>Resultados esperados</b>	El componente “Reporte de los estudiantes de pregrado” brinda toda la información solicitada. Realiza los filtros y exporta el listado de forma satisfactoria.
<b>Evaluación</b>	Prueba satisfactoria

### 3.3.3 Pruebas de Aceptación

Las pruebas de aceptación son creadas en base a 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 éstas 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. [38]

Tabla 34. Caso de Prueba de Aceptación HU35\_P1 para la HU "Generar gráfico de predicción de graduación".

Caso de Prueba de Aceptación	
<b>Código:</b> HU35_P1	Historia de Usuario: No. 5 Generar gráfico de predicción de graduación.
<b>Nombre:</b> Generar gráfico de predicción de graduación general.	
<b>Descripción:</b> Prueba la funcionalidad de mostrar gráfico de predicción de graduación general.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona la opción “Estatística”. El usuario selecciona la opción “Gráfico”. En la opción gráficos de predicción, se selecciona la opción “Geral”.	

Tabla 35. Caso de Prueba de Aceptación de HU15\_P2 para la HU "Filtrar estudiantes de pregrado".

Caso de Prueba de Aceptación	
<b>Código:</b> HU15_P2	Historia de Usuario: No. 15 Filtrar estudiantes de pregrado.
<b>Nombre:</b> Filtrar estudiantes de pregrado.	
<b>Descripción:</b> Prueba la funcionalidad de realizar filtrados en la tabla de estudiantes de pregrado.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona la opción “Relactórios”. El usuario selecciona la opción “Licenciatura”.	

En la opción “Filtrar por”, se seleccionan los filtros que se deseen.
<b>Resultado esperado:</b> Muestra los estudiantes de la tabla según el filtrado que se realice.
<b>Evaluación de la prueba:</b> Prueba satisfactoria.

Tabla 36. Caso de Prueba de Aceptación de HU14\_P3 para la HU "Exportar estudiantes de pregrado".

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU14_P3	Historia de Usuario: No.14: Exportar estudiantes de pregrado.
<b>Nombre:</b> Exportar estudiantes de pregrado.	
<b>Descripción:</b> Prueba la funcionalidad para exportar a Excel los datos de la tabla de estudiantes de pregrado.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona la opción “ <i>Relactórios</i> ”.	
El usuario selecciona la opción “ <i>Licenciatura</i> ”.	
El usuario selecciona la cantidad que desea exportar.	
El usuario selección la opción “Exportar”.	
<b>Resultado esperado:</b> Exporta para Excel la tabla de estudiantes.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

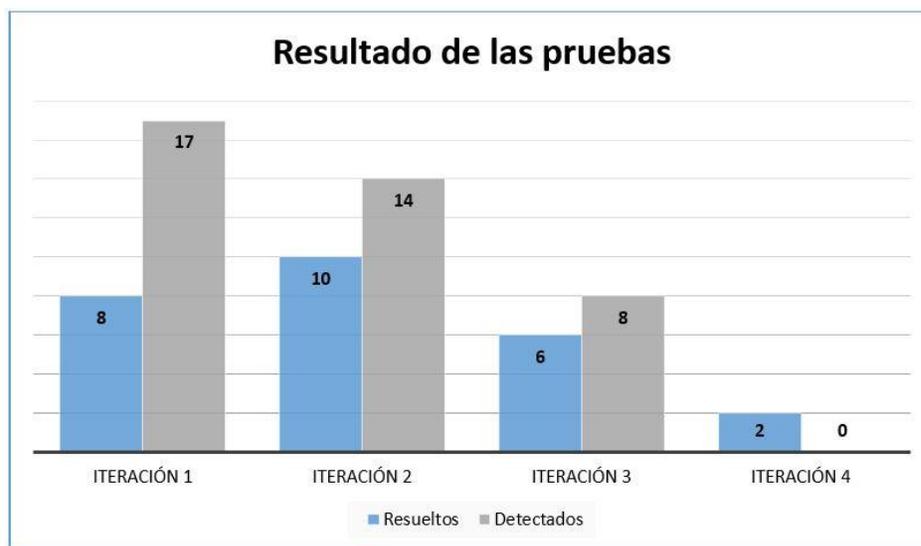
Tabla 37. Caso de Prueba de Aceptación de HU36\_P4 para la HU "Generar gráfico de estudiantes por provincia".

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU36_P4	Historia de Usuario: No. 5 Generar gráfico de estudiantes por provincia.
<b>Nombre:</b> Generar gráfico de estudiantes finalistas por provincia.	
<b>Descripción:</b> Prueba la funcionalidad de mostrar gráfico de estudiantes finalistas por provincia.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona la opción “ <i>Estatística</i> ”.	
El usuario selecciona la opción “ <i>Gráfico</i> ”.	
En la categoría “Estudiantes por provincia”, se selecciona la opción “ <i>Finalista</i> ”.	
<b>Resultado esperado:</b> Muestra un gráfico con las provincias de Angola con la cantidades de estudiantes respectivamente.	

**Evaluación de la prueba:** Prueba satisfactoria.

### 3.3.4 Resultado de las pruebas realizadas

Las pruebas realizadas al módulo reporte estadístico han sido satisfactorias, pues mediante este proceso se han encontrado varias no conformidades en las primeras iteraciones siendo estas resueltas. Finalizando esta fase, se pudo probar que el modulo cumple con todos los requisitos por lo cual fue diseñado. En la figura 14 se muestra la relación de no conformidades detectadas en cada iteración.



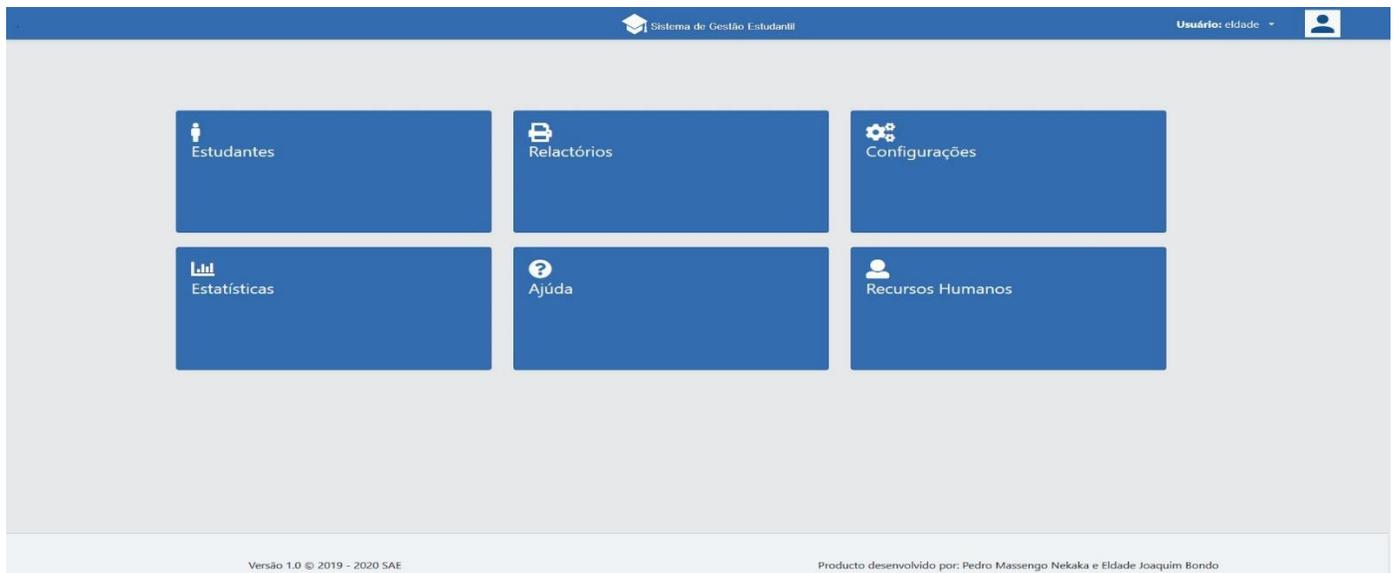
*Figura 14. Resultado de las pruebas por iteración.*

Como se puede ver en la figura anterior se realizaron cuatro iteraciones de pruebas. En la primera iteración se detectaron diecisiete no conformidades, 3 asociadas a errores de interfaz, cuatro errores ortográficos y diez errores de funcionalidad, por lo tanto, se han resuelto ocho (cuatro errores ortográficos, tres de interfaz y uno de funcionalidad). En la segunda iteración se detectaron 14 no conformidades (todos de funcionalidad), se solucionaron diez de ellas. En la tercera iteración se detectaron ocho no conformidades y se resolvieron seis de ellas. En ultima iteración no detecto ninguna no conformidad, por lo tanto, se procedió a la resolución de los dos casos restantes de la iteración anterior, la aplicación mostró un buen funcionamiento y se considera terminada.

### 3.4. Implementación 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. A continuación, se muestran algunas imágenes del sistema implementando.

La interfaz (figura 15) muestra la pantalla principal del Sistema de Gestión Estudiantil. Aquí el usuario de mayor nivel, tiene completo al sistema.



*Figura 15. Página principal de la aplicación*

La interfaz (figura 15) es el módulo de reporte, aquí el usuario realiza búsquedas con filtros y puede exportar la lista a un documento de Excel.

Principal / Relatórios / Licenciatura

Relatório dos estudantes de licenciatura

Nome, Apellidos ...

Filtrar por:

Selecionar a Província de Cuba: 
 Selecionar a Área de Formação: 
 Selecionar o Curso: 
 Selecionar o Ano:

Universidad de las Ciencias Informaticas 
 Selecionar o Genero: 
 Selecionar a Província de Angola:

LISTA GERAL DOS ESTUDANTES DE LICENCIATURA  
 Data: 26/5/2020  
 PAÍS: CUBA

Nº	Nome	Sexo	Ano	Universidade	Curso	Província
1	Adilson Junior da Cruz Cardoso	M	4 *	Universidad de las Ciencias Informaticas	Ciencias Informática	Maianje
2	Anderson Evander Casimiro Aguiar	M	5 *	Universidad de las Ciencias Informaticas	Ciencias Informática	Namibe
3	Armando Lourenço da Silva e Costa	M	5 *	Universidad de las Ciencias Informaticas	Ciencias Informática	Luanda
4	Bento de Assunção Cassule Cusseil	M	3 *	Universidad de las Ciencias Informaticas	Ciencias Informática	Luanda
5	Cléusio de Oliveira Carlos	M	5 *	Universidad de las Ciencias Informaticas	Ciencias Informática	Luanda

Figura 16. Página de reportes de estudantes de pregrado.

La interfaz (figura 16) es el módulo de estadísticas, El usuario visualiza los datos estadísticos a través de ilustraciones gráficas.

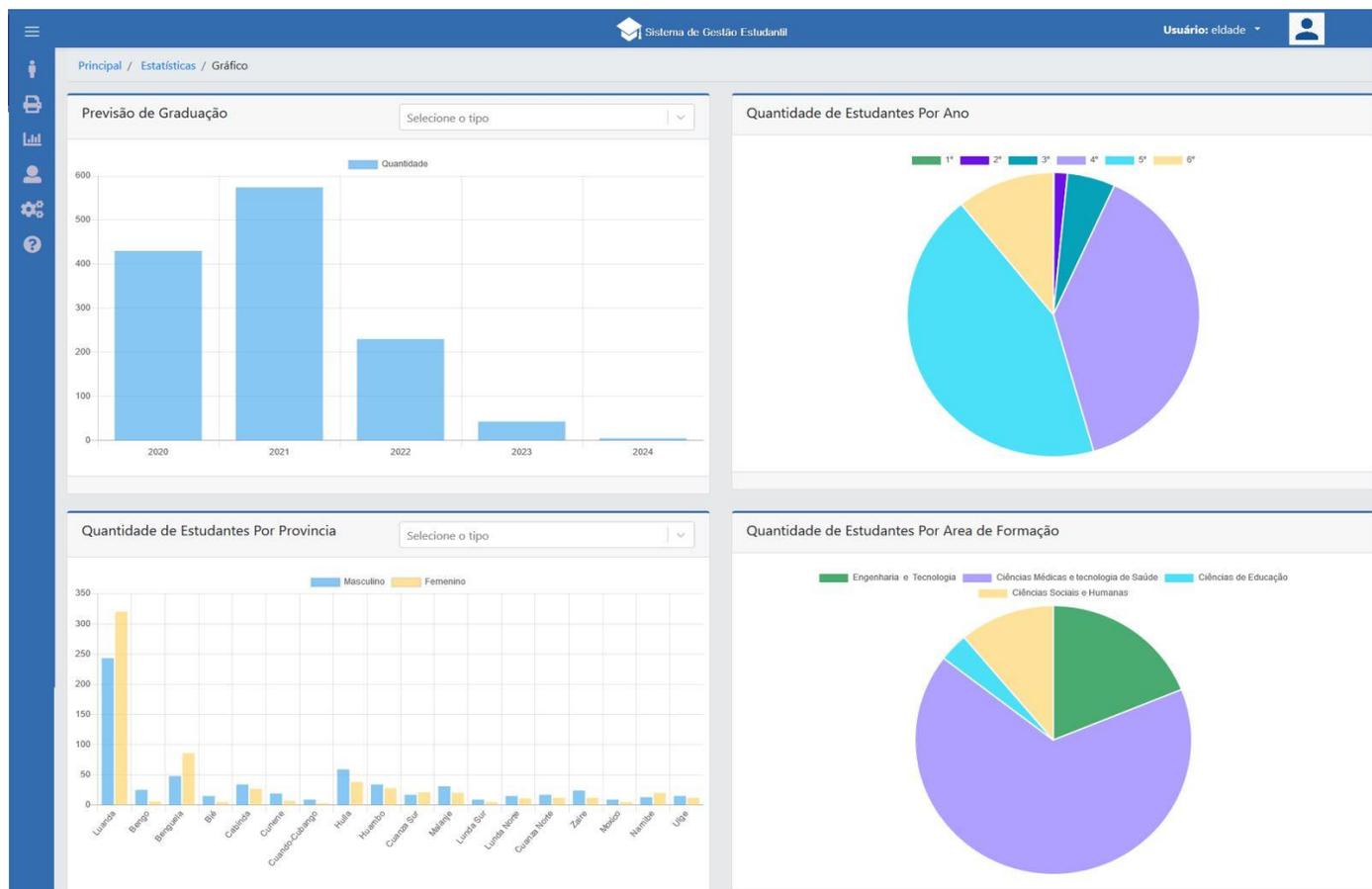


Figura 17. Página de estadísticas por gráficos.

Con la implementación del módulo de reporte estadístico se ha podido solucionar los problemas planteados anteriormente, actualmente con los procesos automatizados, el tiempo de respuesta para las solicitudes de información se ha reducido de forma significativa, se mejoró la precisión de los datos para el pago de los subsidios y la posterior ubicación laboral de los estudiantes una vez terminado su período de formación.

### 3.5. Conclusiones del capítulo

Con el análisis enfocado en la propuesta de solución, se pudo desarrollar el modelo no relacional que fue uno de los principales elementos fundamentales para la implementación del sistema. Con el cumplimiento de las tareas de ingeniería apoyado por las HU se pudo proseguir a las pruebas unitarias y de aceptación que culminaron de forma satisfactoria.

## CONCLUSIONES GENERALES

Terminado el proceso de investigación se concluye lo siguiente:

- ✓ El estudio de diferentes sistemas de gestión estudiantil permitió obtener una idea inmediata de los principales conceptos asociados a la investigación.
- ✓ El levantamiento de requisitos usando Historias de Usuario facilitó una correcta modelación del problema, obteniéndose 38 requisitos funcionales y 12 requisitos no funcionales.
- ✓ Los patrones de diseño identificados se proporcionan al estilo arquitectónico Modelo-Vista-Controlador, ocasionando facilidad en la implementación y mantenimiento del sistema.
- ✓ Con las pruebas de software realizadas al módulo, se garantiza la satisfacción las necesidades y requisitos elaborados por el cliente. Por tanto, el modulo contribuye a mejorar los procesos gestionados por el SAE.

## **RECOMENDACIONES**

El autor realiza las siguientes recomendaciones:

- ✓ Extender el sistema a todos los países con la representación del SAE.
- ✓ Crear una ligación con el sistema implementado en el INAGBE.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] La base de datos líder del mercado para aplicaciones modernas. MongoDB [en línea], [sin fecha]. [Consulta: 16 octubre 2019]. Disponible en: <https://www.mongodb.com/es>.
- [2] 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.
- [3] BHANDARI, D., 2019. Job Vacancy Application using MERN stack. [en línea]. [Consulta: 4 diciembre 2019]. Disponible en: <http://www.theseus.fi/handle/10024/170363>.
- [4] React – Una biblioteca de JavaScript para construir interfaces de usuario. [en línea], [sin fecha]. [Consulta: 16 octubre 2019]. Disponible en: <https://es.reactjs.org/>.
- [5] NODE.JS, [sin fecha]. Node.js. Node.js [en línea]. [Consulta: 16 octubre 2019]. Disponible en: <https://nodejs.org/es/>.
- [6] 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>.
- [7] LINDÉN, L., 2019. AngularJS-sovelluksen päivittäminen Angular-sovellukseksi. [en línea]. [Consulta: 4 diciembre 2019]. Disponible en: <http://www.theseus.fi/handle/10024/260986>.
- [8] 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>.
- [9] 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>.
- [10] PETUKHOVA, E., 2019. Sitecore JavaScript Services Framework Comparison.
- [11] GAUCHAT, J.D., 2012. El gran libro de HTML5, CSS3 y JavaScript. S.I.: Marcombo. ISBN 978-84-267-1782-5.
- [12] La base de datos líder del mercado para aplicaciones modernas. MongoDB [en línea], [sin fecha]. [Consulta: 16 octubre 2019]. Disponible en: <https://www.mongodb.com/es>.

- [13] Documentation for Visual Studio Code. [en línea], [sin fecha]. [Consulta: 16 octubre 2019]. Disponible en: <https://code.visualstudio.com/docs>.
- [14] MEZA-CALDERÓN, D.C., 2018. Modelación hidrológica de la cuenca media del Río Abangares en Guanacaste. [en línea], [Consulta: 4 diciembre 2019]. Disponible en: <https://repositoriotec.tec.ac.cr/handle/2238/9888>.
- [15] 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>.
- [16] A. Justiz Munder, Luis. C. Peñalver Peñalver, José. Directorio v2.0 de la Universidad de las Ciencias Informáticas.2016.
- [17] I. B. Núñez, «REFLEXIÓN BIBLIOGRÁFICA SOBRE SISTEMAS INFORMÁTICOS PARA LA GESTIÓN DE INFORMACIÓN DE MAESTRÍAS EN CENTROS UNIVERSITARIOS», *Universidad&Ciencia*, vol. 6, n.º 3, pp. 26-35, oct. 2017.
- [18] NIEVA CHAVES, J.A. y MARTÍNEZ CHACÓN, O., 2017. Una nueva perspectiva de la formación docente en el Servicio Nacional de Aprendizaje (SENA), Colombia. *Revista Cubana de Educación Superior*, vol. 36, no. 1, pp. 109-119. ISSN 0257-4314.
- [19] 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>.
- [20] 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>.
- [21] Abraham SilbSillberschatz y Henry F. Korth, *Database System Concepts*, Sexta. McGraw-Hill, 2015.
- [22] Faiza Anwer, Shabib Aftab, y Iftikhar Ali, 2017. Proposal of Tailored Extreme Programming Model for Small Projects.
- [23] LETELIER, P. y PENADÉS, M.C., 2006. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [www.cyta.com.ar/ta0502/v5n2a1.htm](http://www.cyta.com.ar/ta0502/v5n2a1.htm) [en línea]. [Consulta: 23 febrero 2020]. Disponible en: <https://www.cyta.com.ar/ta0502/v5n2a1.htm>.

[24] 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>.

[25] (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](https://www.researchgate.net/publication/299506242_METODOLOGIAS_TRADICIONALES_VS_METODOLOGIAS_AGILES).

[26] IAN, Sommerville. 2006. Ingeniería del software. México DF: Pearson. 2005, 2006.

[27] 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%20208/UML%20y%20Patrones%20%202da%20Edicion.pdf>

[28] Giraldo G., Gloria L.; Acevedo O., 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>

[29] 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](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>.

[30] 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>

[31] Abraham SilbSillberschatz y Henry F. Korth, Database System Concepts, Sexta. McGraw-Hill, 2015

[32] 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/>

[33] Data Modeling Introduction. [en línea]. [Consulta: 11 mayo 2020]. Disponible en: <https://docs.mongodb.com/manual/core/data-modeling-introduction/>

[34] Data Model Design. [en línea]. [Consulta: 11 mayo 2020]. Disponible en: <https://docs.mongodb.com/manual/core/data-model-design/>

[35] 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>

[36] Andy Hernández Paez. Ingeniería de Software aplicada a los trabajos de diploma. Facultad 4. Universidad de las Ciencias Informáticas.

[37] Roger S. Pressman. Ingeniería del Software un enfoque práctico. Séptima edición. [Capítulo 17: Estrategias de Prueba de software]

[38] José Joskowicz. Reglas y Practicas en eXtreme Programming. [en línea]. [Consulta: 16 mayo 2020]. Disponible en: [https://www.academia.edu/download/31398587/xp\\_-\\_jose\\_joskowicz.pdf](https://www.academia.edu/download/31398587/xp_-_jose_joskowicz.pdf)

[39] «¿Qué es HTML5? | HTML5», 14-nov-2018. [En línea]. Disponible en: <https://www.arkaitzgarro.com/html5/capitulo-1.html>. [Consulta: 12 diciembre 2019].

[40] B. R. M. Roger S. Pressman, Software Engineering. A Practitioner’s Approach, 8th ed. New York: McGraw-Hill Education, 2015.

## **ANEXOS**

### **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. ¿Qué sistema se usa actualmente la gestión de la información?
2. ¿Qué limitaciones presenta el sistema actual?
3. ¿De qué manera las limitaciones influyen en el buen funcionamiento de la institución?
4. ¿Cómo se garantiza la seguridad del sistema actual?
5. ¿Cómo se conservan los datos de los estudiantes que no están en activo?
6. ¿Qué dificultades hay en elaborar los reportes estadísticos?
7. ¿Con que fin se elaboran los reportes estadísticos?
8. ¿Qué tipo de reportes se deben generar?

#### **Entrevista realizada a:**

Director del SAE: Dr. Eugenio Novais



REPÚBLICA DE ANGOLA  
MINISTÉRIO DO ENSINO SUPERIOR  
INSTITUTO NACIONAL DE GESTÃO DE BOLSAS DE ESTUDO

DECLARAÇÃO DE COMPROMISSO

Eu Eldade Joaquin Bando ao tomar  
Conhecimento de parte dos Deveres e Direitos constantes no Regulamento das Bolsas de Estudo  
Externas, comprometo-me a seguir o curso de Engenharia Informática  
a nível de Licenciatura No País Cuba

DEVERES DO BOLSEIRO

- Ter comportamento moral e cívico irrepreensível;
- Ter bom comportamento académico;
- Apresentar anualmente ao INAGBE/MES a documentação comprovativa do aproveitamento académico;
- Prestar todas as declarações e informações que lhe forem solicitadas pelas entidades competentes;
- Não mudar de Instituição de ensino nem do curso sem autorização prévia do INAGBE, sob pena de corte da bolsa de estudo;
- Não abandonar o curso ou estágio antes de o completar, salvo por motivo considerado válido pelos órgãos que atribuíram a bolsa de estudo.
- Cumprir o regulamento do bolseiro e outras disposições que forem aplicáveis
- Dar conhecimento de todas as circunstâncias que possam afectar o normal rendimento académico;
- Velar pela manutenção dos bens matérias postas à sua disposição;
- Não viajar para terceiros países sem prévia autorização dos órgãos o que atribuíram a bolsa de estudo.

DIREITOS DO BOLSEIRO

- Usufruto da bolsa de estudo;
- Auxílio que possa vir a obter nos termos da legislação;
- Viajem de ida e volta paga pelo Estado, no início e no fim da sua formação.

Comprometo-me ainda, a cumprir os demais regulamentos e normas vigentes

**Obs:** Custear todos encargos caso queira desistir da formação abandonando o País acolhedor

Luanda, 06 de Outubro de 2014

Estudante bolseiro

Eldade J. Bando

Os Funcionário

Manoel Baptista  
Instrutor do processo

DGN.-

O Chefe de Departamento

Figura 18. Declaração de compromisso del estudiante becario.