
Desarrollo de una herramienta de extracción, almacenamiento y procesamiento de datos para el monitoreo de egresados del programa de Ingeniería de Sistemas de la Corporación Universitaria del Caribe CECAR

Jose David Ballesteros Paternina

Corporación Universitaria del Caribe – CECAR
Facultad de Ciencias Básicas, Ingeniería y Arquitectura
Programa de Ingeniería de Sistemas
Sincelejo
2021

Desarrollo de una herramienta de extracción, almacenamiento y procesamiento de datos para el monitoreo de egresados del programa de Ingeniería de Sistemas de la Corporación Universitaria del Caribe CECAR

Jose David Ballesteros Paternina

Trabajo de grado presentado como requisito para optar al Título de Ingeniería de Sistemas

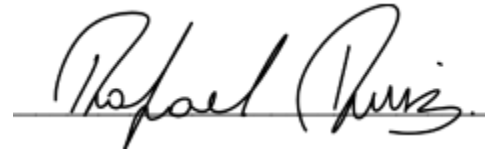
Director

Mg. Rafael Roberto Ruiz Escorcía

Corporación Universitaria del Caribe – CECAR
Facultad de Ciencias Básicas, Ingeniería y Arquitectura
Programa de Ingeniería de Sistemas
Sincelejo
2021

Nota de Aceptación

4,23



Director



Evaluador 1



Evaluador 2

Sincelejo, Sucre, 11 de octubre de 2021

Dedicatoria

A Dios porque siempre me dio sabiduría y entendimiento para poder llevar a cabo este proyecto y de igual forma porque mis padres me enseñaron que siempre hay que agradecerle todo a él.

A mi Mamá por el apoyo y motivación dado en todo este tiempo.

A mi novia por siempre estar para mí y por todo este tiempo que me apoyó desde el comienzo del desarrollo de este proyecto de grado, animándome y motivándome.

A mi profesor y director de tesis Rafael Roberto Ruiz Escorcía por la calidez brindada en la formación académica y en el apoyo del proyecto en cada momento.

Jose David Ballesteros Paternina

Agradecimientos

Quiero agradecer a todas las personas que hicieron parte de este proceso, al cuerpo docente que acompañó durante la carrera, especialmente al Docente Rafael Roberto Ruiz Escorcía quien fue mi tutor y director durante el desarrollo de este proyecto, por su apoyo teórico y práctico, nos brindó, asesorías, tiempo y gestiones que hicieron posible facilitar muchos procesos necesarios para cumplir nuestro desarrollo de este proyecto, de igual manera a la Corporación Universitaria del Caribe por el apoyo y brindarme la formación de alta calidad para la culminación de mi carrera profesional.

Tabla de Contenido

Resumen.....	13
Abstract	14
Introducción	15
1. Diseño técnico y metodológico	18
1.1. Modelo de ciclo de vida del software.....	18
1.1.1. Modelo en cascada.....	19
1.1.1.1. Análisis.....	20
1.1.1.2. Diseño.....	21
1.1.1.3. Codificación.....	21
1.1.1.4. Pruebas o validaciones.....	22
1.1.1.5. Mantenimiento.....	22
1.2. Metodología de desarrollo de software	22
1.2.1. Extreme Programming o XP.....	23
1.2.1.1. Planificación del proyecto.....	24
1.2.1.2. Diseño.....	25
1.2.1.3. Codificación y pruebas.....	25
1.2.2. Justificación de la metodología seleccionada para el proyecto	25
2. Caracterización de los procesos de recolección de datos de la Corporación Universitaria del Caribe para e monitoreo y seguimiento de los graduados del programa de Ingeniería de Sistemas.....	26
2.1. Problemática.....	27
3. Implementación de la metodología.....	28
3.1. Diseño de la herramienta.....	30
3.2. Herramientas de desarrollo.....	33
3.3. Web Scraping	34
3.4. Justificación de las tecnologías implementadas en el proyecto	36
3.5. Codificación y pruebas de la herramienta.....	39

3.5.1.	Detalles de ejecución y pruebas de los Scripts	40
3.5.1.1.	Configuración de Selenium WebDriver	42
3.5.1.2.	Inicialización Navegador y sitio web a Screapear	44
3.5.1.3.	Acceder a cada perfil del resultado de búsqueda.....	46
3.5.1.4.	Validación si la persona ya se encuentra almacenada en el archivo JSON.....	51
3.5.1.5.	Extracción de datos en la sección de Experiencia	53
3.5.1.6.	Navegación entre páginas de búsqueda de Google.....	58
3.5.1.7.	Búsqueda por medio de diferentes palabras claves	60
3.5.1.8.	Creación de archivo JSON.....	62
3.5.1.9.	Clasificación de los datos	63
4.	Resultados.....	71
4.1.	Tiempo que tardan los Estudiantes del programa de Ingeniería de Sistemas en graduarse 73	
4.2.	Perfiles que tienden a certificarse los graduados del programa de Ingeniería de Sistemas 74	
4.3.	Perfiles que tienden a laborar los graduados del programa de Ingeniería de Sistemas..	76
4.4.	Instituciones que tienden a elegir los graduados del programa de Ingeniería de Sistemas para continuar su evolución educativa	77
4.5.	Ubicaciones de trabajo de los graduados de Ingeniería de Sistemas	79
4.6.	Perfiles que tienden a elegir los graduados para continuar con su proceso de formación en el campo de la ingeniería de sistemas.....	81
5.	Resultados Registro Calificado del Programa de Ingeniería de Sistema del año 2017	83
5.1.	Salario.....	83
5.2.	Medios de obtención de empleos	84
5.3.	Principales roles o cargos desempeñados.....	85
5.4.	Sectores económicos de desempeño	86
6.	Conclusiones.....	89
7.	Recomendaciones	91
	Referencias Bibliográficas	93
	Anexos	96

Lista de Figuras

Figura 1. Fases del modelo en cascada	19
Figura 2. Diagrama arquitectónico del sistema	29
Figura 3. Diagrama de clases	30
Figura 4. Diagrama de paquetes	31
Figura 5. Prueba de caja negra	36
Figura 6. Configuración de Selenium WebDriver	37
Figura 7. Inicialización navegador y sitio web a Scrapear	38
Figura 8. Inicialización del driver y el navegador web	39
Figura 9. Navegador en ejecución	40
Figura 10. Sección educación de perfil a scrapear	42
Figura 11. Perfil que no cumple con los requisitos a scrapear	42
Figura 12. Valida si el perfil cuenta con las validaciones especificadas	43
Figura 13. Valida si el perfil cuenta con las validaciones especificadas	43
Figura 14. Valida si el perfil cuenta con las validaciones especificadas	44
Figura 15. Valida si el perfil cuenta con las validaciones especificadas	45
Figura 16. Validación si ya se encuentre en el archivo JSON	46
Figura 17. Bloque de código el cual extrae la sección de experiencias	48
Figura 18. Bloque de código que extra la sección de educación	49
Figura 19. Bloque de código que extra la sección de certificaciones	50
Figura 20. Guardar datos en el archivo JSON	50
Figura 21. Ejemplo estructura de un bloque de datos almacenados en JSON	51
Figura 22. Sección Experiencia	51
Figura 23. Sección Educación	52

Figura 24. Sección certificaciones	52
Figura 25. Páginas de resultados de Google	53
Figura 26. acciones de dar clic en las diferentes páginas de resultados de Google	54
Figura 27. Bloque de código de búsqueda por medio de diferentes palabras claves	55
Figura 28. Bloque de código de búsqueda por medio de diferentes palabras claves	56
Figura 29. Bloque de código de búsqueda por medio de diferentes palabras claves	56
Figura 30. Bloque de código que crea, elimina, actualiza el archivo JSON	57
Figura 31. Bloque de código de clasificación de datos scrapeados	59
Figura 32. Bloque de código de clasificación de datos scrapeados	60
Figura 33. Bloque de código de clasificación de datos scrapeados	60
Figura 34. Bloque de código de clasificación de datos scrapeados	61
Figura 35. Bloque de código de clasificación de datos scrapeados	62
Figura 36. Bloque de código de clasificación de datos scrapeados	62
Figura 37. Bloque de código de clasificación de datos scrapeados	63
Figura 38. Bloque de código de clasificación de datos scrapeados	63
Figura 39. Bloque de código de clasificación de datos scrapeados	64
Figura 40. Bloque de código de clasificación de datos scrapeados	65
Figura 41. Bloque de código de clasificación de datos scrapeados	66
Figura 42. Recuento por años en que se gradúan los estudiantes del programa de Ingeniería de Sistemas	69
Figura 43. recuento por años en que se gradúan los estudiantes	70

Figura 44. perfiles que tienden a certificarse los graduados del programa de Ingeniería de Sistemas	72
Figura 45. Gráfico de torta de perfiles en lo que tienden a laborar los graduados del programa de Ingeniería de Sistemas	74
Figura 46. Gráfico de torta que muestra el resultado de las instituciones que tienden a elegir los graduados del programa de Ingeniería de Sistemas para continuar su evolución educativa	75
Figura 47. Mapa donde se visualiza las ciudades y/o municipios donde actualmente desempeñan su carrera profesional o han laborado los graduados del programa de Ingeniería de Sistemas...	76
Figura 48. Mapa donde se visualiza los países en los cuales actualmente desempeñan su carrera profesional o han laborado los graduados del programa de Ingeniería de Sistemas	77
Figura 49. Salario Ingreso Graduados Por Área.....	86
Figura 50. Medios Principales para la Obtención de Empleos.....	86
Figura 51. Principales roles o cargos desempeñados por los graduados del programa de Ingeniería de Sistemas.....	87
Figura 52. Sectores Económicos Ubicación graduados	88

Lista de tablas

Tabla 1. Herramientas implementadas en el proyecto.....	32
Tabla 2. Realización de prueba 1	38
Tabla 3. Realización de prueba 2	41
Tabla 4. Realización de prueba 3	45
Tabla 5. Realización de prueba 4	47
Tabla 6. Realización de prueba 5	53
Tabla 7. Realización de prueba 6	55
Tabla 8. Realización de prueba 7	57
Tabla 9. Realización de prueba 8	58
Tabla 10. Realización de prueba 9	66
Tabla 11. Resultados del Tiempo que tardan los estudiantes del programa de Ingeniería de sistemas en graduarse	68
Tabla 12. Resultados de los perfiles que tienden a certificarse los graduados del programa de Ingeniería de sistemas	70
Tabla 13. Resultados de los perfiles que tienden a laborar los graduados del programa de Ingeniería de Sistemas	72
Tabla 14. Resultados de las Instituciones que tienden a elegir los graduados del programa de Ingeniería de sistemas para continuar con su evolución educativa	73
Tabla 15. Resultados de las locaciones en las cuales estan o han estado desempeñando laboralmente su profesión los graduados del programa de Ingeniería de Sistemas	74
Tabla 16. Recuento de países en los que están ejerciendo o han ejercido su perfil profesional los graduados del programa de Ingeniería de Sistemas	76
Tabla 17. resultados de recuento de los perfiles que tienden a elegir los gradudos para continuar con su proceso de formacion en el campo de la ingenieria de Sistema	77
Tabla 18. Comparación de Salarios Graduados CECAR	84

Anexos

Anexos 1. Reunión con la Coordinadora MARIA ANGELICA GARCIA MEDINA donde se abarcó las necesidades de la Corporación.....98

Anexo 2. Archivo Excel de egresados de Ingeniería de Sistemas.....99

Anexo 3. Código fuente de la herramienta desarrollada100

Anexo 4. Informe de análisis de web scraping en Power BI101

Resumen

La acreditación de programas académicos en las universidades es muy importante para obtener la certificación de los diferentes programas que hacen parte de su oferta académica, es por esto que es indispensable contar con una herramienta de extracción de datos que ayude a llevar un monitoreo de la evolución educativa que han tenido los egresados después de haber culminado su carrera profesional. La Corporación Universitaria del Caribe CECAR es una institución de educación superior acreditada en diversos programas apoyando así la educación a nivel regional. Dicha institución cuenta con una técnica de recolección de datos realizada por medio de entrevistas enviadas por correo electrónico y llamadas telefónicas, no obstante una problemática evidente en la aplicación de dichos métodos es la poca participación por parte de los estudiantes egresados debido a que optan, en su mayoría, por desvincularse de la institución haciendo caso omiso al llamado y al ofrecimiento de servicios por parte de la institución que conlleven a una recolección de datos relacionados con su desarrollo educativo y laboral. Durante el desarrollo de este proyecto se hizo una revisión a los estudios de seguimiento a egresados mediante el uso de análisis multivariados por medio de técnicas de recolección de datos como Web Scraping. Con este proyecto se buscó la toma de decisiones por parte de la institución entorno al mejoramiento de sus procesos, agilizar tiempos de gestión en la obtención de la información, facilitar al acceso de la información, aumento de la eficiencia en la gestión de los procesos y demás, lo que traduce como un beneficio sustancial para la institución.

Palabras clave: acreditación, web scraping, extracción de datos, análisis multivariados, procesos.

Abstract

The accreditation of academic programs in universities is very important to obtain the certification of the different programs that are part of their academic offerings, which is why it is essential to have a data extraction tool that helps to monitor the educational evolution of graduates after completing their professional career. The Corporación Universitaria del Caribe CECAR is a higher education institution accredited in various programs, thus supporting education at the regional level. This institution has a technique of data collection by means of interviews sent by e-mail and telephone calls, however, an evident problem in the application of these methods is the low participation on the part of the graduated students due to the fact that most of them choose to disengage from the institution, ignoring the call and the offer of services by the institution that lead to the collection of data related to their educational and labor development. During the development of this project, a review of graduate follow-up studies was made through the use of multivariate analysis by means of data collection techniques such as Web Scraping. With this project, the institution sought to make decisions to improve its processes, speed up management times in obtaining information, facilitate access to information, increase efficiency in the management of processes and others, which translates as a substantial benefit for the institution.

Keywords: accreditation, web scraping, data extraction, data processing, multivariate analysis, process.

Introducción

El monitoreo de egresados es un procedimiento realizado en su mayor parte por instituciones de educación superior que desean conocer y analizar la calidad educativa que le fue brindada a los estudiantes. Asimismo, la aplicación de este ejercicio permite a las instituciones “contar con una herramienta útil para la toma de decisiones con respecto a un nuevo diseño, revisión, modificación o actualización de planes y programas de estudio” (Velázquez, 2010).

Llevar a cabo un seguimiento sistemático de los egresados brinda la oportunidad de innovar los procesos de planeación, ejecución y evaluación de cada uno de los programas con los que cuenta la corporación. Por otra parte, la aplicación de nuevos métodos de enseñanza permite aumentar el grado de competitividad de los egresados con relación a la formación recibida, la cual garantice el acceso al mercado laboral incrementando el índice de empleabilidad, para lograr estos objetivos es necesario conocer las necesidades actuales y aspiraciones de los futuros egresados.

En este sentido, “el egresado es una fuente importante de retroalimentación, en tanto que permite a la universidad conocer dónde y cómo está ubicado, su rol social y económico y la forma de reflejar los valores adquiridos durante su formación académica” (Morales et al., 2008). Ahora bien, el seguimiento a egresados resulta ser una pieza fundamental para las instituciones de educación superior ya que se encuentra dentro de los requisitos necesarios para la acreditación de alta calidad académica emitida por el Ministerio de Educación.

Dicho lo anterior, el seguimiento de egresados debe ser contemplado como una estrategia que le permitiría a la Corporación contar con la acreditación de todos sus programas, es por esto, que contar con la opinión de los egresados facilitaría el cumplimiento de requisitos requeridos por *el Consejo Nacional de Acreditación (CNA) y el Sistema Nacional de Acreditación (SNA)*. “El Sistema Nacional de Acreditación, SNA es el conjunto de políticas, estrategias, procesos y organismos cuyo objetivo fundamental es garantizar a la sociedad que las instituciones de educación superior que hacen parte del sistema cumplen con los más altos requisitos de calidad y que realizan sus propósitos y objetivos”. (Artículo 53 de la Ley 30 de 1992).

Para la Corporación Universitaria del Caribe CECAR, el proceso de acreditación de los programas educativos es muy importante ya que ayuda a mejorar la calidad con la que presta el servicio de educación superior y obtener así la acreditación de alta calidad para cada uno de los programas que este brinda. Ahora bien, para alcanzar una acreditación de alta calidad, es necesario contar con diferentes requisitos la cual una de ellas es medir la tasa de graduación por periodo y de igual manera llevar un monitoreo de la evolución educativa y laboral que han tenido los egresados luego de haber culminado su carrera profesional, para así obtener un análisis de estos datos y así facilitar la toma de decisiones enfocada al mejoramiento educativo.

Actualmente, la Corporación Universitaria del Caribe CECAR lleva a cabo un seguimiento de sus egresados a través de la realización de encuestas diseñadas para recolectar información de sus graduados acerca de su desarrollo en el mercado laboral y la relación que existe entre su profesión con el trabajo que están desempeñando. Esta técnica lleva consigo una deficiencia en la obtención de datos, debido a que los graduados al momento de finalizar su proceso formativo en la Institución se desvinculan totalmente de esta lo que lleva a una deficiencia en la recolección de datos y al difícil análisis de estas.

Debido a esto, la solución para la problemática con la que cuenta la corporación Universitaria del Caribe CECAR al recolectar información sobre sus egresados, es desarrollar una herramienta que sea capaz de obtener datos de los graduados del programa de Ingeniería de Sistemas que sea óptimo y ayude al monitoreo de su evolución a nivel académico y laboral después de haber culminado su carrera profesional.

Es por esto, que la intención de este proyecto fue, desarrollar una herramienta para la extracción, almacenamiento y procesamiento de datos desde el aplicativo web LinkedIn que permita obtener un análisis exploratorio de los datos de los egresados del programa de Ingeniería de Sistemas de la Corporación Universitaria del Caribe CECAR que se encuentran registrados en esta plataforma, con el fin de realizar un seguimiento a su evolución académica y laboral después de culminar su formación profesional.

Para lograr este objetivo, se recurrió primeramente a identificar los requerimientos necesarios para el desarrollo de la herramienta de extracción, almacenamiento y procesamientos de datos. Seguidamente se procedió a analizar diferentes herramientas, tecnologías y métodos de extracción, almacenamiento y procesamiento de datos de la plataforma LinkedIn para efectuar una caracterización de los egresados del programa de ingeniería de sistemas. A partir del análisis, se construyó la herramienta que ayudó a la extracción, almacenamiento y procesamiento de datos para el seguimiento a la evolución académica y laboral de los egresados del programa de ingeniería de sistemas después de culminar su formación académica, y finalmente se realizaron pruebas unitarias que permitieron asegurar el buen funcionamiento de la herramienta y así suministrar datos aplicables a un posterior análisis de estas.

Este proyecto se fundamenta en facilitar y apoyar los procesos de seguimiento a los graduados, basándose en los datos obtenidos mediante el registro calificado del año 2017 del programa de Ingeniería de sistemas de la Corporación Universitaria del Caribe realizado por esta misma institución, dando a conocer información detallada de la evolución académica de los graduados, la cual fue obtenida por medio de la realización de encuestas por parte de la Institución.

A través de estas encuestas, se pudo evidenciar los diferentes datos que fueron extraídos por la Institución para el estudio de la evolución académica y profesional de los graduados. Esto favoreció a tener una visión más clara de los datos a extraer por medio de la herramienta desarrollada, como lo fue la evolución académica. Esto hace referencia, a la educación y certificaciones que han obtenido los graduados después de haber culminado su carrera profesional y/o extracurriculares. De igual forma a los perfiles, ciudad y/o países en los cuales los graduados tienden a ejercer su profesión.

Es por esto que, este proyecto busca apoyar el proceso que realiza la Institución para la recolección de datos por medio de encuestas a los graduados y así contar con mayores datos reales que ayuden a realizar un análisis más profundo de estas.

1. Diseño técnico y metodológico

En esta sección se evidencia el trabajo realizado en la ejecución del proyecto en cuanto a la estructuración, modelos, metodologías y demás herramientas que fueron usadas durante todo el proceso de creación. Para este caso en particular el modelo de ciclo de vida del software usado fue el modelo en cascada. Para la metodología de desarrollo se optó por la metodología de desarrollo ágil XP. A continuación, se describen las razones por las cuales cada herramienta, metodología y tecnologías escogidas se adaptan perfectamente a la construcción de la herramienta que se iba a desarrollar, del mismo modo se evidencia los resultados obtenidos por medio del proceso de extracción y procesamientos de datos, donde se implementaron diferentes análisis que permitieron dar valor a estos datos para así transformarlo en información significativa que ayude a la Corporación Universitaria del Caribe a tener una mejor visión y tomar mejores decisiones hacia la población estudiantil del programa de Ingeniería de Sistemas.

1.1. Modelo de ciclo de vida del software

El ciclo de vida en un proyecto empieza cuando se da la recolección de requerimientos para el programa que se va a desarrollar, así mismo, termina cuando el programa se completa y es entregado. Sin embargo, durante el desarrollo del programa surgen diversas fases por las cuales se tiene que pasar, es por esto que la metodología empleada debe cumplir con varias etapas las cuales están compuestas por cada uno de los modelos que hacen parte de ciclo de vida del software.

"Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso." (Organización Internacional de Normalización, [ISO], 12207-1)

Es por esto que, se hace necesario garantizar y verificar que el software cumpla con todos los requisitos para su aplicación, para lo cual se debe asegurar que la metodología empleada es la apropiada.

Llegado a este punto del proyecto, se evidenciará toda la estructura que se llevó a cabo durante el proceso de desarrollo de la herramienta, la cual abarca todo lo relacionado con el modelo

de desarrollo implementado, el cual fue el modelo en cascada y de igual forma la metodología utilizada, la cual fue la Metodología Ágil XP donde, a continuación, se evidenciará todo el seguimiento meticuloso de las diferentes fases a seguir, sumado a todas las herramientas y tecnologías utilizados.

1.1.1. Modelo en cascada.

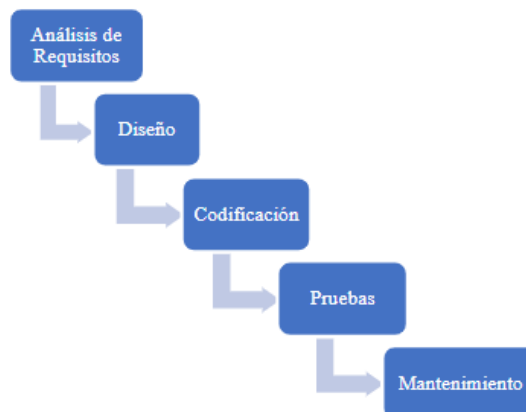
El modelo en cascada se define como una secuencia de pasos, que al finalizar cada etapa reúne toda la documentación para asegurar que cumple con los requisitos y especificaciones. Este modelo para la época se convirtió en una base fundamental de ejemplo de proceso dirigido, donde se planificaría todas las actividades antes de empezar a trabajar en ellas.

“Este modelo propone un enfoque secuencial y sistemático para el desarrollo de software, conlleva más disciplina y se basa principalmente en las etapas de análisis de requisitos, diseño, codificación, pruebas y mantenimiento.” (Sommerville, 2006).

A continuación, se puede observar gráficamente las fases que contiene este modelo de desarrollo en cascada.

Figura 1

Fases del modelo en cascada



Fuente: Pressman, Roger (2010)

El modelo en cascada es uno de los modelos más utilizados en lo que respecta al ciclo de vida de un software, este modelo se describe como un modelo lineal, el cual consta de varias fases que se deben seguir y completar para llegar a la siguiente. Estas fases son:

1.1.1.1. Análisis.

En esta etapa de desarrollo, se establece la visión del problema, desde el punto de vista del cliente y de igual forma desde la parte de los desarrolladores, donde se especifica el alcance que tendrá el proyecto y de igual forma contempla las funcionalidades que contendrá el software o producto.

“Independientemente de lo bien diseñado o codificado que esté un programa, si se ha analizado y especificado pobremente, decepcionará al usuario y desprestigiará al que lo ha desarrollado”. (Palacios, 2006).

Es por esto que, para construir un sistema lo más importante es comprender el contexto y la necesidad del cliente o usuario final, y a su vez, esta etapa es la más compleja de realizar. El anterior autor citado, afirma que, si se ha realizado de manera incorrecta esta fase del ciclo de desarrollo de software, el resultado final fallará o tendrá inconsistencias, lo cual, se convierte en un proceso difícil de corregir en otras etapas más avanzadas o en el peor de los casos al final del ciclo de desarrollo de software. Debido a esto, es importante realizar un levantamiento de los requerimientos de forma correcta teniendo en cuenta las diferentes técnicas de recolección de estas como entrevistas, encuestas, investigación, entre otros.

1.1.1.2. Diseño.

Mientras que en la etapa de análisis se realiza el levantamiento de requerimientos, en esta fase de diseño se estudia la forma de implementar los requerimientos entregados por el usuario final o cliente, es aquí donde se especifica el cómo se hará.

De igual forma que en la etapa de análisis, esta contiene diferentes manera de visualizar estos requerimientos de manera gráfica con ayuda a describir cómo el software o producto va a satisfacer los requerimientos, donde se especifica desde cada pieza que tendrá el software, los subsistemas que este contendrá, cada pieza de software o hardware complementario que serán necesarios para su mejor rendimiento, la arquitectura del software, hasta cada tecnologías, herramientas y lenguajes de programación que ayudarán a cumplir cada requerimiento funcionales como no funcionales acordado con el usuario final.

1.1.1.3. Codificación.

Luego de realizar el proceso de diseño de la arquitectura del software, se ejecuta la fase de codificación, en la que se incluye la construcción del software, la búsqueda de errores, y pruebas unitarias. En esta fase el proyecto de software se traduce al lenguaje de programación que fue posteriormente descrito en la etapa de diseño. Los diferentes componentes se desarrollan por separado, teniendo en cuenta las diferentes técnicas de codificación y patrones de diseño que se vayan a implementar.

Esta fase de codificación, da como resultado el producto de software que se comprueba por primera vez como producto final en la fase de pruebas y validaciones.

1.1.1.4. Pruebas o validaciones.

En esta fase de prueba incorpora la integración del software en el entorno seleccionado, en donde se someterá a diferentes técnicas de pruebas, desde caja negra, caja blanca, pruebas de performance entre otros. Las pruebas de aceptación desarrolladas en la fase de análisis permiten determinar si el software cumple con todas las especificaciones y requerimientos definidos anteriormente en la etapa de análisis. Al momento de superar esta etapa con éxito, este ya está listo para su lanzamiento o puesta en producción para su uso.

1.1.1.5. Mantenimiento.

Una vez finalizada la fase de pruebas y validaciones de manera exitosa, se autoriza la aplicación productiva del software. Esta última etapa del modelo en cascada incluye la entrega del producto, desde la documentación de los requerimientos funcionales y no funcionales, como en la codificación. También se realiza la entrega del código fuente del software al cliente, y posteriormente al mantenimiento y mejora del software.

Como se puede observar, el ciclo de vida de un software elaborado bajo este modelo debe estar bien estructurado, ya que si se incumplen algunas de las fases anteriormente mencionadas es imposible lograr que el software avance, debido a que cada una de estas fases se ejecuta una vez.

1.2. Metodología de desarrollo de software

“Lo más importante en el desarrollo de software en equipo es la comunicación. Cuando un problema surge en el desarrollo, la mayoría de las veces alguien ya conoce la solución, pero ese conocimiento no llega a alguien con el poder de hacer el cambio.” (Beck, 2017).

La metodología de desarrollo de software es el entorno de trabajo utilizado para planificar, estructurar y controlar los procesos de desarrollo de un software, con el objetivo de trabajar de manera regulada.

El entorno de trabajo de una metodología de desarrollo de software, consiste en la aplicación de técnicas y métodos por medio de los cuales se pueden diseñar o desarrollar productos o soluciones. En el desarrollo de software, la metodología hace énfasis en el funcionamiento de procesos organizativos a través del cual se llega al desarrollo de un sistema, reduciendo su nivel de complejidad y agilizando su proceso de crecimiento.

Es por esto que, hoy en día es indispensable el uso de metodologías ágiles para el desarrollo de proyectos tecnológicos debido a que estos traen muchos beneficios, entre las cuales se destaca

por la entrega del producto en desarrollo en un menor tiempo, mejor organización entre en equipo de trabajo y de igual forma la asignación de tareas para cada equipo que hagan parte del proceso de creación de proyectos tecnológicos.

Cabe destacar que existen diferentes metodologías ágiles que se acomodan a diferentes maneras de trabajo en equipo, como SCRUM, XP, Kanban entre otros. Para este proyecto se consideró el uso de la metodología de desarrollo ágil XP o Extreme Programming debido a su simplicidad, facilidad para adaptarse a cualquier situación que presente problemas y hasta procesos más técnicos como la reutilización del código desarrollado.

1.2.1. Extreme Programming o XP

La Programación Extrema, es un enfoque de la ingeniería de software formulado por Kent Beck, “se considera el más destacado de los procesos ágiles de desarrollo de software. Al igual que estos, se diferencia de los métodos tradicionales principalmente en que presenta más énfasis en la adaptabilidad que en la previsibilidad.” (Bautista Q, 2012).

Este tipo de metodología ágil se diferencia de otras debido a que propone principalmente la adaptabilidad antes que la previsibilidad, es decir, es enfocada más en la organización y planificación para que no haya errores durante todo el proceso de desarrollo y así realizar iteraciones más satisfactorias y eficientes. De igual manera, esta metodología puede ser flexible y ser implementada para un solo programador, Según (Agarwal y Umphress, 2008) definen una tabla de procesos todas aquellas actividades que deben ser ejecutadas para adaptar la metodología XP a un desarrollo realizado por un programador. En esta tabla se contemplan tres tareas principales las cuales son de: planeación, desarrollo y Post Mortem, donde esta última se refiere al proceso de completar las pruebas de aceptación del código en la línea de base de producción y de igual manera, cada una de estas tareas contienen actividades específicas que permitirán al programador aplicar el concepto de PXP (Personal extreme Programming).

Como toda metodología ágil, Extreme Programming, también cuenta con diferentes fases tomando como base la definición planteada por (Bustamante y Rodríguez, 2014), donde exponen las siguientes fases de la metodología XP:

1.2.1.1. Planificación del proyecto.

Esta metodología plantea la planificación como un diálogo continuo entre el cliente, los programadores y los coordinadores del proyecto. Donde se empieza por la recolección de información con ayuda de diferentes métodos. Al tener bien estructuradas las necesidades del cliente, se evalúa el tiempo estimado por cada requerimiento.

El resultado de la fase de planificación del proyecto fue:

- Entrevista.

1.2.1.2. Diseño.

En esta fase es importante tener en cuenta la planificación y requerimientos, ya que ayudará al proceso de construcción de los diferentes diagramas y la elaboración formal de la arquitectura que contará la herramienta, estableciendo las tecnologías adecuadas para la solución del problema.

Los resultados de la fase de diseño fueron:

- Arquitectura del software.
- Diagrama de clases.
- Diagrama de paquetes.

1.2.1.3. Codificación y pruebas.

En esta fase, se implementará el diseño realizado, satisfaciendo cada requerimiento. De igual forma se realizará diferentes pruebas que ayuden a validar y verificar el buen funcionamiento de la herramienta, desde la extracción de los datos, su comportamiento y salidas de esta misma.

Los resultados de la fase de codificación fueron:

- Código fuente de la herramienta.

1.2.2. Justificación de la metodología seleccionada para el proyecto

La metodología aplicada cuenta con múltiples ventajas que da soporte a su implementación dentro del proyecto. Una de esas ventajas es su flexibilidad de ser modificada para trabajar con un solo individuo. XP permite que el flujo de los procesos sea cambiado antes de llegar a una fase más avanzada y así mitigar problemas a largo plazo. Por otra parte, Según Iyawa (2016) es menos costoso adoptar XP en comparación con las metodologías tradicionales metodologías porque hay menos cantidad de retrabajo involucrado cuando se aplica XP.

Asimismo, la metodología presenta algunas desventajas como dificultad para documentar ya que este trabaja de forma rápida y con cambios constantes, no obstante, este no afectó en el proceso de desarrollo del proyecto, debido a que desde la fase de planificación se contó con el alcance y visión del desarrollo de la solución. Por otra parte, esta metodología cuenta con otra desventaja la cual es la fuerte dependencia de las personas, pero gracias al contar con una ventaja la cual es “la flexibilidad con respecto a los requisitos del cliente” (Qureshi & Ikram, 2015). Es por esto que se puede optar por realizar modificaciones como la Personal Extreme Programming la cual fue mencionada anteriormente.

Al realizar este proyecto de manera individual, es conveniente implementar el concepto Personal Extreme Programming (PXP). Por lo que, Agarwal y Umphress escalaron la metodología XP para producir una metodología que utilice sus prácticas, pero de una forma en la que un individuo dentro del marco de un proyecto tradicional pueda utilizar. “Las prácticas de XP se modifican para que puedan encajar en una situación de programador solitario y se crea un proceso de desarrollo de software. Llamamos a nuestro método PXP (Personal Extreme Programming).” (Agarwal y Umphress 2008).

Por consiguiente, esta metodología fue la más adaptable para el desarrollo de este proyecto ya que, por su funcionalidad, condiciones, ventajas y técnicas de implementar PXP (Personal Extreme Programming) ayuda a llevar el proceso desde la planificación del proyecto hasta la

implementación de una forma más fácil y flexible, simplemente contando con una sola persona en el proceso de desarrollo del proyecto.

2. Caracterización de los procesos de recolección de datos de la Corporación Universitaria del Caribe para e monitoreo y seguimiento de los graduados del programa de Ingeniería de Sistemas

Para realizar la caracterización de los procesos de recolección de datos de la Corporación Universitaria del Caribe, inicialmente, se identificaron las problemáticas que se encuentran actualmente en el proceso de recolección de datos, con el fin de desarrollar unos objetivos que ayudaran a dar respuesta a la problemática que presenta la institución.

Seguidamente, se realizó una reunión el día 7 de octubre de 2020 con la Ingeniera Maria Angelica Garcia Medina, la cual, en esta fecha contaba con el cargo como Coordinadora del programa, donde se conversó de las diferentes dudas y problemáticas, que presenta la institución hasta el día de hoy en el proceso de recolección de datos para el monitoreo de los graduados del programa de Ingeniería de Sistema. El proceso que lleva la institución para la recolección de datos de los graduados es la siguiente:

La Coordinación de Graduados envía un formulario llamado Estudio De Seguimiento E Impacto por medio de correo electrónico. Este cuenta con un grupo de correos de los graduados, que por consiguiente diligencian este formulario. Este se compone de tres (3) secciones que los graduados deben diligenciar, estos son: Características sociodemográficas, empleabilidad del graduado y pertinencia e impacto social.

- **Características sociodemográficas**

En esta sección se encuentran consignados rasgos demográficos y de hogar que se indagan a la población de graduados.

- **Empleabilidad del graduado**

Esta sección indaga aspectos de la dinámica laboral del graduado, así como las características y competencias necesarias para el desarrollo profesional y personal dentro del mercado de trabajo.

- **Pertinencia e impacto social**

En esta sección se pretende identificar la dinámica del graduado desde la pertinencia de la formación y el impacto en el desarrollo social y económico a partir de la diferenciación que marcan los graduados como parte sustancial y como la expresión del trabajo desarrollado por estos en la sociedad que dan cuenta de la calidad del programa que los formó.

2.1. Problemática

La Corporación Universitaria del Caribe es una institución de educación superior, ubicada en la carretera Troncal de Occidente Km. 1 vía Corozal - Sincelejo, Colombia. Durante la charla con la Coordinadora del programa de Ingeniería a cargo en la fecha establecida anteriormente, se pudo evidenciar las necesidades con las que cuenta hasta el día de hoy la institución en el proceso de recolección de datos, y escasez de estas mismas con respecto a los estudiantes graduados. Es por esto, que se pudo establecer la problemática que se afronta la institución al momento de monitorizar la evolución académica y laboral de los graduados del programa de Ingeniería de Sistemas.

- Técnica de recolección de datos ineficiente.
- Pocos datos recolectados, ya que debido al tamaño de la muestra conlleva a un bajo porcentaje de fiabilidad.
- Proceso de monitoreo manual.

Pero antes de empezar con el diseño de esta herramienta, fue necesario requerir el listado de los egresados hasta la actualidad. Este documento fue necesario solicitarlo por medio de correo electrónico al Decano de la Facultad de Ingeniería y Arquitectura, que en la fecha contaba con ese

cargo el Ingeniero Guillermo Carlos Hernandez Hernandez, es cual muy amablemente nos facilitó esta valiosa información.

Referidas las problemáticas, el objetivo principal a desarrollar para dar una solución a las problemáticas encontradas, fue el desarrollo de una herramienta de extracción, de datos de los graduados de la institución, específicamente del programa de Ingeniería de Sistemas para el monitoreo de la evolución académica y laboral con ayuda del sitio web LinkedIn. Seguidamente se realizaría el procesamiento de datos para transformarlo en información y darle significado a esta. Luego se almacenará toda esta información con el fin de contar con un historial que ayude a tener un mayor porcentaje de datos reales para así realizar un análisis de estas con ayuda de herramientas que ayuden a la visualización por medio de gráficas para un mayor entendimiento.

3. Implementación de la metodología

Como se había mencionado anteriormente, la metodología con la que se desarrolló este proyecto fue XP, sin embargo, el proyecto no contaba con la cantidad necesaria de personas para dividir cada integrante y asignar tareas o actividades que la metodología plantea. Es por esto que se tomó la modificación de esta metodología para una sola persona llamada Personal Extreme Programming que en la sección de la metodología se estuvo comentando su funcionamiento, ya que este proyecto se trabajó con un solo individuo o persona. Debido a esto, el proyecto fue realizado de manera exitosa aplicando buenas prácticas y trayendo consigo grandes resultados, pero sobre todo muchas ideas para desarrollos futuros con respecto a esta misma problemática que se está atacando. Lo primero que se realizó fue la recolección de información, aquí se establecieron las necesidades y problemáticas de la Corporación Universitaria Del Caribe - CECAR con respecto al monitoreo de los egresados del programa de Ingeniería de Sistema, y continuando con la fase de desarrollo donde se aborda el análisis, diseño y pruebas de la herramienta y sus diferentes Scripts a realizar de manera que este cumpla con el alcance y objetivo deseado, una herramienta que pueda extraer los datos y procesarlos para su debido análisis.

Cabe destacar que, al haber culminado el proyecto, se obtuvo como resultado una herramienta que extracción de datos acorde a las necesidades de la Institución que ayudó a aumentar la cantidad de datos de 138 graduados de 329 actualmente (Anexo 2), dando así un 39,51% de datos reales obtenidos. Se obtuvieron este número de graduados ya que, un número de egresados, más precisamente un 5% (17 egresados) contaban con una cuenta en LinkedIn, pero esta no presentando datos que diera valor a la toma de decisiones.

A continuación, se describirán el diseño arquitectónico, diagrama de componentes y diagrama de clases de la herramienta. Del mismo modo se describirán las tecnologías que fueron utilizadas para este y la razón porque era la mejor opción, también se mostrará el proceso de la implementación de la metodología XP con la modificación a PXP y finalmente se mostrará los resultados obtenidos de los datos extraído por medio del sitio web LinkedIn.

3.1. Diseño de la herramienta

Luego de haber analizado la problemática proporcionada por la Ingeniera Maria Angelica Garcia Medina y establecido el alcance, se establece como consecuencia de estos procesos el diseño de la herramienta.

Figura 2

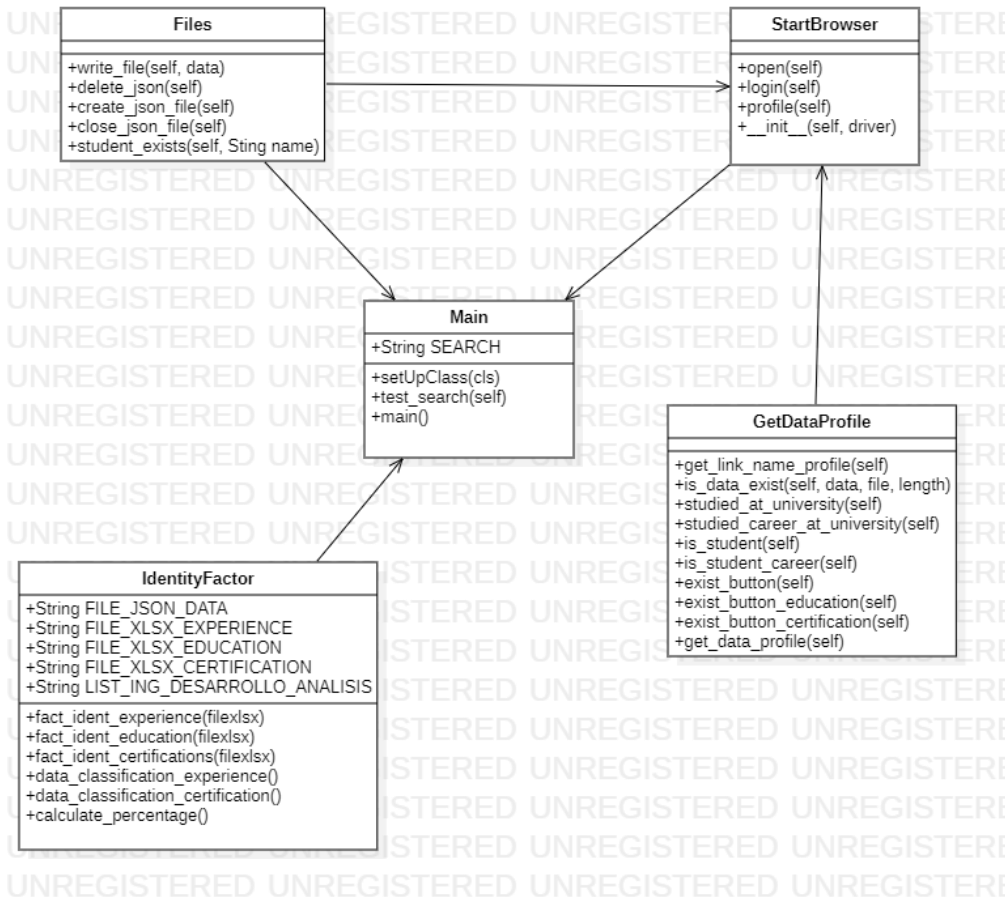
Diagrama arquitectónico del sistema



Fuente: Elaboración propio

Figura 3.

Diagrama de clases

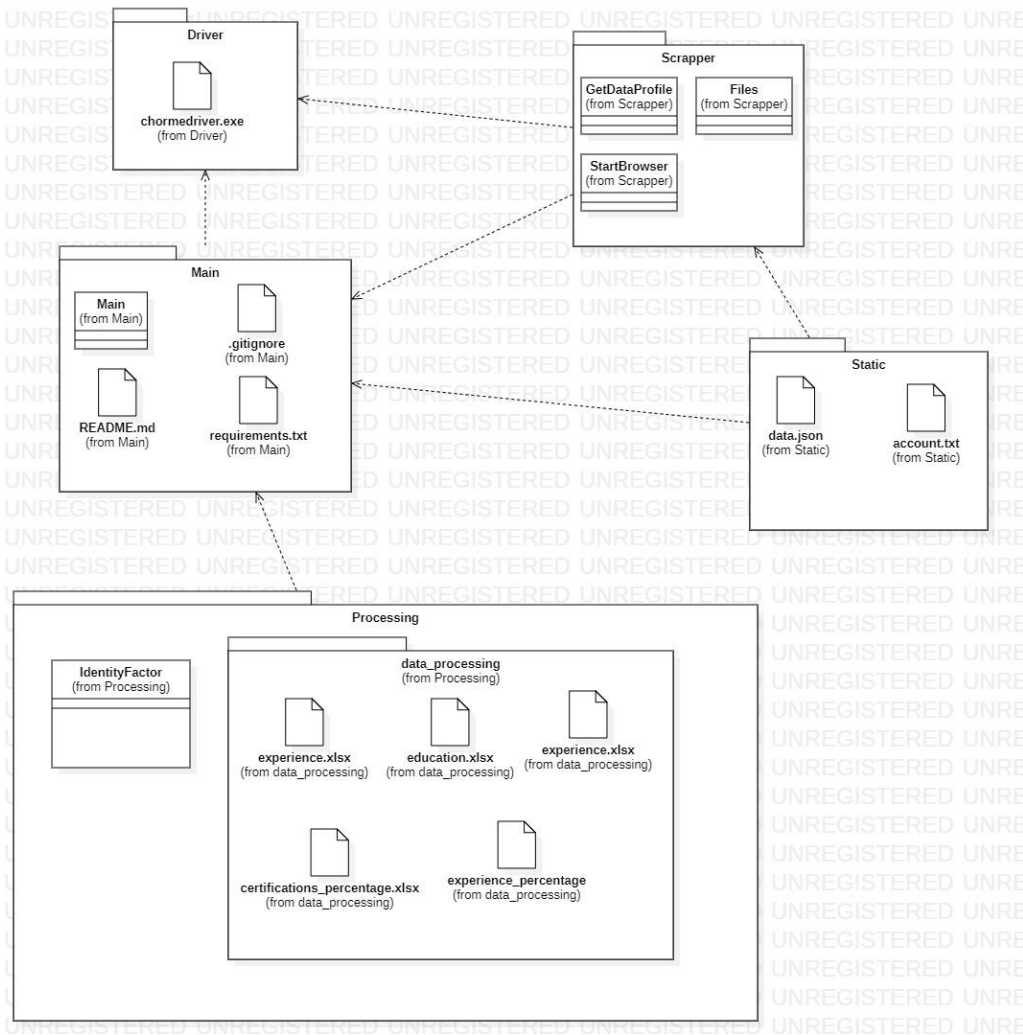


Fuente: Elaboración propio.

El diagrama de clases ayuda a representar los elementos que componen una herramienta o sistema desde un punto de vista estatico, es decir, facilita el mejor entendimiento de como se encuentra estructurado un proyecto de software desde la parte de codificación, donde se evidencia las diferentes clases o archivos de codigo con sus respectivos atributos y metodos que este se compone y de igual manera el impacto y la importancia que tiene cada uno de estos archivo. Este diagrama ayuda a tener una mejor visualización y mayor claridad de como se encuentra estructurada la herramienta.

Figura 4.

Diagrama de paquetes



Fuente: Elaboración propio

El diagrama de paquetes ayuda a definir y visualizar los distintos paquetes a nivel lógicos que forman parte de la herramienta y la dependencia entre ellos, dando una mayor claridad de como se encuentra empaquetado cada archivo de código.

3.2. Herramientas de desarrollo

Tabla 1

Herramientas de desarrollo

Herramienta	Versión	Descripción
Python	3.9	Lenguaje de programación para la creación de Scripts para la extracción de datos y procesamiento de esta.
Selenium Web Driver	ChromeDriver 92.0.4515.107	Herramienta de código abierto utilizada para automatizar procesos realizados en los navegadores web y de igual forma para extracción de información de la web.
Json	6.1	Formato de texto sencillo para el intercambio de datos estructurados
Firebase	5.0.2	Base de datos no relacional para la persistencia de datos
Pycharm	Community Edition 2020.3.3	Entorno de desarrollo integrado.
Microsoft Excel	2019	Editor de hoja de cálculos para la conexión de datos con Power BI
Power BI	2.96.1061.0	Sistema de conexión de datos, modelados y visualización de datos por medio de diferentes gráficas
Git	2.32.0	Control de versiones descentralizado
GitHub	3.1.6	Sistema de control de versiones
Windows	Windows 10 Pro 64 bit	Sistema operativo donde se llevó a cabo la ejecución de los diferentes Scripts

Fuente: Elaboración propia

3.3. Web Scraping

Los datos son una parte esencial de cualquier investigación, ya sea académica, de marketing o científica. Es posible que las personas quieran recopilar y analizar datos de varios sitios web. Los diferentes sitios web en categorías específicas muestran información en diferentes formatos. Incluso con un solo sitio, es posible que no pueda ver todos los datos al mismo tiempo. Los datos se pueden distribuir en varias páginas en diferentes secciones. “La mayoría de los sitios web no le permiten guardar una copia de los datos que se muestran en sus páginas web en el almacenamiento local”. (Penman et al., 2009).

Es aquí donde el Web Scraping ayuda a extraer datos significativos de una o varias páginas web determinadas, para una manipulación o análisis posterior. Existen herramientas que nos permiten extraer datos web, también podemos crear programas en lenguajes de programación como Python, Java, Ruby entre otros.

Pero, ¿el Web Scraping es legal? Esta técnica de recolección de datos es completamente legal como técnica informática que es. Hoy por hoy, muchas empresas están implementando esta técnica de web Scraping. Se habla que el 45% del tráfico en la Web es movida por robots y no por personas. Es casi tan antiguo como la propia Internet; ya que sus primeros usos fue el de organizar la información que se encontraba en esta para posteriormente indexarla.

Esta técnica durante mucho tiempo la toman como ciberdelincuencia porque es una técnica que ha sido utilizada en numerosas ocasiones con fines ilícitos. Como muchas otras herramientas y técnicas, puede ser ilegal dependiendo del uso que se haga de ella. Por ejemplo, el correo electrónico. ¿Es ilegal mandar spam?, Por supuesto. ¿Eso convierte al email en ilegal? Claro que no.

El scraping se utiliza de forma ilegal tanto desde el punto de vista del uso: “Para qué utilizamos esta información” como de la técnica; “Cómo conseguimos esta información”. Uno de los usos que hace que esta técnica sea ilegal es la copia de material para utilizarlo como propio, ya que estamos violando los derechos de autor. Este es un claro ejemplo de ilegalidad desde el punto de vista de “cómo utilizamos esta información”. Un ejemplo del mal uso en la forma de obtener información con el web scraping es la siguiente: Deseamos obtener el listado de precios de una

tienda online, donde la finalidad en este caso es legal ya que son datos públicos, pero el programador que creó el bot o robot no contaba con la experiencia, buenas prácticas, ni conocimientos sólidos con respecto a este tema. Su robot visita la tienda para obtener estos datos de precios mandando muchas peticiones a este sitio web, las cuales el servidor del web objetivo es incapaz de dar respuesta y este deja de funcionar durante un tiempo con la consecuente pérdida de ingresos que esto supone para el sitio. Sin pensarlo, se ha hecho un ataque de denegación de servicios o DDoS.

Es por esto, que en el proceso de construcción de esta herramienta se tuvo en cuenta cada situación que pudiese afectar a la plataforma de LinkedIn, donde se tomaron las mejores prácticas y se realizó una ardua investigación y prácticas para evitar todo inconveniente que pudiera afectar a la plataforma de LinkedIn. Se tuvo presente el número de peticiones al sitio web a scrapear, donde se llegó al desarrollo de una herramienta que no afecta el rendimiento de esta misma, y llevándolo a lo más exacto a las acciones que realiza una persona dentro de este sitio web.

Ahora bien, ¿en qué casos es ilegal el scraping? Como se mencionó, el scraping no supone una ilegalidad en sí mismo. Pero su uso para ciertos fines puede ser ilegal. Si queremos mantenernos en la legalidad debemos estar atentos y usarla de manera ética, teniendo especial cuidado de no usar propiedad intelectual o marcas comerciales, no violar los derechos de autor, no practicar la competencia desleal. Y, por supuesto, no sobrecargar los servidores de los sitios eliminados. Siendo esto tomado como referencia e implementada para el desarrollo de esta herramienta, cumpliendo con cada una de estas reglas.

3.4. Justificación de las tecnologías implementadas en el proyecto

Las tecnologías y herramientas sobre las cuales se desarrolló el proyecto fueron: Python, librería de Python como Pandas, Selenium, Firebase y Power BI. La facilidad y la constante actualizaciones de estas tecnologías son las principales características tenidas en cuenta al momento de escogerlas, siendo Python un lenguaje de programación versátil, con una fácil curva de aprendizaje, su potencia en el manejo y procesamiento de datos con sus diferentes librerías como Pandas. Es uno de los lenguajes con una gran comunidad que ayuda cada día en su mejoramiento y realización de nuevas versiones. De igual forma, Python es un lenguaje de programación elegante y flexible, donde no es necesario preocuparse tanto por los detalles, es ordenado y limpio, donde su sintaxis y su indentación hace que sea fácil de leer y entender, es un lenguaje muy portable en todos los Sistemas Operativos existentes hasta el día de hoy en comparación con otros lenguajes de programación.

Por otra parte, Selenium no es una sola herramienta, sino un Suite de software, cada uno con un enfoque diferente para apoyar diferentes problemáticas, como pruebas de automatización, extracción de datos, entre otros. Está formado por cuatro componentes principales que incluyen:

- Entorno de desarrollo Integrado Selenium (IDE)
- Selenium Remote Control
- WebDriver
- Selenium Grid

En este caso en específico, nos centraremos en Selenium WebDriver ya que este fue el componente que se utilizó para atacar la problemática que cuenta la Institución. Esta es la herramienta más importante del Suite de Selenium.

Selenium WebDriver proporciona una interfaz de programación para crear y ejecutar casos de prueba. Los Script de prueba se escriben con el fin de identificar los elementos web en las páginas web y luego se realizan las acciones deseadas en esos elementos. Pero más allá de poder realizar pruebas o acciones que realiza un ser humano dentro de un navegador web, este también

ayuda a extraer datos de cualquier Sitio Web el cual ayuda a realizar Web Scraping fácilmente desde Python.

Por otra parte, luego de tener claro las herramientas y tecnologías para el proceso de Web Scraping, es importante tener presente que herramienta es la indicada y la cual tenga mejor rendimiento con Python, es por esto que se optó por utilizar Pandas. Una librería muy popular de Python, sobre todo en el ámbito de Data Science, ya que ofrece una gran estructura poderosa y flexible que facilita la manipulación y tratamiento de datos.

Data Science es la ciencia que se centra en el estudio de los datos, se encarga de obtener una inmensa cantidad de datos, donde esta se combina con la estadística, las matemáticas y la informática para interpretar datos la cual lleva a la toma de decisiones. Estos datos se extraen mediante diferentes canales como teléfonos móviles, redes sociales, e-commerce, encuestas o sitios webs, donde estas son algunas de las fuentes utilizadas.

Esta ciencia cuenta con una serie de conceptos básicos los cuales tiene diferentes campos donde se puede especializar para solucionar diferentes problemas. Estos conceptos claves son los siguientes:

- **Data Mining:** Es el proceso utilizado para la recolección y almacenamiento de datos útiles, por lo que es necesario analizar patrones de datos en grandes cantidades usando software. Gracias a este proceso, las empresas pueden extraer más información sobre sus clientes y desarrollar estrategias que favorezcan a sus procesos. Este ayuda a la toma de mejores decisiones con base a la información obtenida.
- **Deep Learning:** Este resuelve problemas a través de redes neuronales los cuales imitan el comportamiento cerebral. Estas se estructuran en capas donde cada una de estas se cuentan con diferentes responsabilidades, donde la primera capa se capta la información y estos datos pasan a la siguiente capa la cual se encarga de realizar diferentes cálculos y finalmente la información recopilada se proyecta en la última de las capas.
- **Machine Learning:** Esta se encarga de educar a la tecnología para que corrija errores por si sola. Se encarga de realizar predicciones y clasificaciones de datos para obtener información útil aplicable a diferentes áreas.

- **Inteligencia Artificial o IA:** Se encarga de utilizar algoritmos para la creación de máquinas que imitan el comportamiento humano.

Hoy en día, el mundo se mueve alrededor de los datos, donde este cuenta con un gran impacto para las organizaciones, empresas, y entidades para la toma de decisiones. Es por esto que es necesario contar con la persistencia de datos ya que esta tiene la capacidad de guardar datos o información de un programa para posteriormente volver a utilizarlas en cualquier momento que sea necesario. Es por esto que, para el proceso de almacenamiento, se optó por Firebase. Es una plataforma ubicada en la nube desarrollada por Google, la cual usa un conjunto de herramientas para la creación y sincronización de proyectos, brindando la posibilidad de almacenar datos.

A diferencia de MySQL y PostgreSQL, los cuales son gestores de bases de datos relacionales, Firebase es un gestor de base de datos no relacional, esto quiere decir que están diseñadas específicamente para modelado de datos específicos y tiene un esquema flexible para crear aplicaciones modernas. Estas son ampliamente reconocidas debido a su simplicidad y facilidad de desarrollar, por su funcionalidad y el rendimiento a escala. Por otro lado, estas bases de datos NoSQL o no relacionales se almacenan como un documento JSON, a diferencia de las bases de datos relacionales, estas están construidas por medio de tablas relacionadas entre sí. Debido a esto, se eligió Firebase, al estar trabajando con documentos JSON, facilitó el trabajo para llevar la persistencia de los datos extraídos desde el sitio web LinkedIn.

Por último y más importante la herramienta usada para la visualización y obtención de los resultados de los datos extraídos, fue Power BI, esta herramienta es una solución de análisis que permite unir diferentes fuentes de datos con el fin de analizarlos y presentar un análisis de esto ya se por medio de informes o paneles. Power BI cuenta con una manera fácil de acceso a datos en cualquier dispositivo, como tablets, laptops y smartphones por lo que se puede disponer de la información en tiempo real. Esta herramienta permite conectar datos de distintas maneras, desde la nube o hasta entornos locales, creando informes o gráficas que faciliten la visualización de los datos. El acceso a datos puede ser desde archivos Excel, CSV, JSON hasta formatos más complejos como lo son las bases de datos. Debido a esto se optó por esta potente herramienta, el cual facilitó la visualización de los datos dando una visión más amplia de la evolución e inclinaciones de los

estudiantes graduados del programa de Ingeniería de Sistema de la CORPORACIÓN UNIVERSITARIA DEL CARIBE.

3.5. Codificación y pruebas de la herramienta

En esta sección se abordará sobre el proceso de codificación de la herramienta de extracción de datos, cómo funciona, y de igual manera las acciones que esta hace. También se abordará la técnica que se optó para realizar las diferentes pruebas de cada uno de los scripts de la herramienta. Cabe destacar que, esta herramienta actúa de igual forma de cómo una persona realiza acciones dentro de un navegador web, desde abrir el navegador ya sea Google Chrome, Firefox, Opera, hasta proceder a interactuar con este con sus diferentes elementos como botones, barra de búsqueda, Ingreso de datos, inicio de sesiones dentro de sitios webs y entre otros.

Antes que nada, en este proyecto se hablará sobre Scripts, los cuales son documentos que contienen instrucciones escritas en un lenguaje de programación, que en esta ocasión están contruidos sobre Python, los cuales ejecutan diversas funciones o acciones necesarias con diferentes herramientas o software.

Ahora bien, la técnica que se utilizó para el proceso de pruebas fue la técnica de caja negra la cual se enfoca a que el sistema realice las operaciones esperadas, es decir, está no conoce el cómo se hace ni el rendimiento, sino que con la información de entrada se obtenga un resultado en la salida esperada. Esta técnica consiste en la construcción de casos de pruebas por cada entrada o acción posible sin importar su validez.

Figura 5.

Técnica de prueba de caja negra



Fuente: pmoinformatica.com

3.5.1. Detalles de ejecución y pruebas de los Scripts.

A continuación, se realizará un resumen del funcionamiento de los diferentes Scripts desarrollados para el proceso de extracción, y procesamiento de datos de los graduados del programa de Ingeniería de sistemas desde la plataforma de LinkedIn.

Para tener en cuenta, los datos que fueron extraídos de la página de LinkedIn fueron los siguientes:

- Nombre completo del usuario
- URL del perfil.
- Sección de Experiencia
- Sección de Educación
- Sección de Certificados y licencias

3.5.1.1. Configuración de Selenium WebDriver.

Primeramente, se procede a configurar el WebDriver, ya que este se encargará de realizar las acciones que hace una persona al momento de interactuar con un sitio web. Esta configuración trata de especificar a qué Sitio Web vamos a realizar el respectivo Web Scraping. En este caso será el sitio web LinkedIn.

Figura 6.

Configuracion Selenium WebDriver

```
from selenium.common.exceptions import NoSuchElementException
from time import sleep
from get_data_profile import GetDataProfile
from files import Files

# Test branch
class StartBrowser:
    def __init__(self, driver):
        self._driver = driver
        self._files = Files()

# call the url
def open(self):
    self._driver.get('https://www.linkedin.com')
```

Fuente: Elaboración propia

Seguidamente se procede a configurar en qué navegador se va a realizar el proceso de extracción y de igual forma se especifica dónde se encuentra alojado el driver de Selenium dentro de nuestro proyecto. También se procede a importar todas las librerías que utilizaremos en todo el proceso de construcción de los scripts, desde la librería para realizar pruebas hasta la herramienta de Selenium.

Figura 7.

Configuración Selenium WebDriver.

```
import unittest
from selenium import webdriver
from start_browser import StartBrowser
from files import Files

SEARCH = 'site:linkedin.com/in/ AND "ingenieria de sistemas" AND "Corporacion Universitaria del Caribe"'

class GoogleTest(unittest.TestCase):

    @classmethod
    def setUpClass(cls):
        cls.driver = webdriver.Chrome(executable_path=r'drivers/chromedriver.exe')
        cls.driver.maximize_window()
```

Fuente: Elaboración propia.

Realización de prueba

Tabla 2.

Ejecución de pruebas

Pasos de ejecución	<ol style="list-style-type: none"> 1. Abrir el navegador. 2. Entrar al sitio web LinkedIn
Resultado esperado	<ol style="list-style-type: none"> 1. Carga del navegador y sitio web LinkedIn correctamente
Resultado de evaluación de la prueba	Exitosa / OK

Fuente: Elaboración propia

3.5.1.2. Inicializar Navegador y sitio web a Scrapear.

Como anteriormente se había comentado que, Selenium realiza las acciones de una persona dentro de un sitio web, se procede a especificar todas las acciones o pasos para entrar al sitio web de LinkedIn.

Primeramente, es necesario cargar LinkedIn y posteriormente ingresar las credenciales de la cuenta de LinkedIn. En este caso de forma educativa, se utilizó una cuenta secundaria la cual tengo registrada en este sitio web. Luego es redireccionado a Google, donde aquí se empezará a realizar la búsqueda de los graduados por medio de funcionalidades que presenta Google, el cual se puede especificar exactamente qué se quiere buscar. En este caso se utilizó la siguiente regla:

site: linkedin.com/in/ AND "ingeniería de sistemas" AND "Corporación Universitaria del Caribe"

Esta especifica que realizará una búsqueda de los perfiles en LinkedIn que cuenten con diferentes palabras claves como Ingeniería de sistemas y Corporación Universitaria del Caribe. Esto hará un match con todos los perfiles que cuenten con estas palabras claves o se asemejen a esta.

Figura 8.

Inicialización del driver y el navegador web

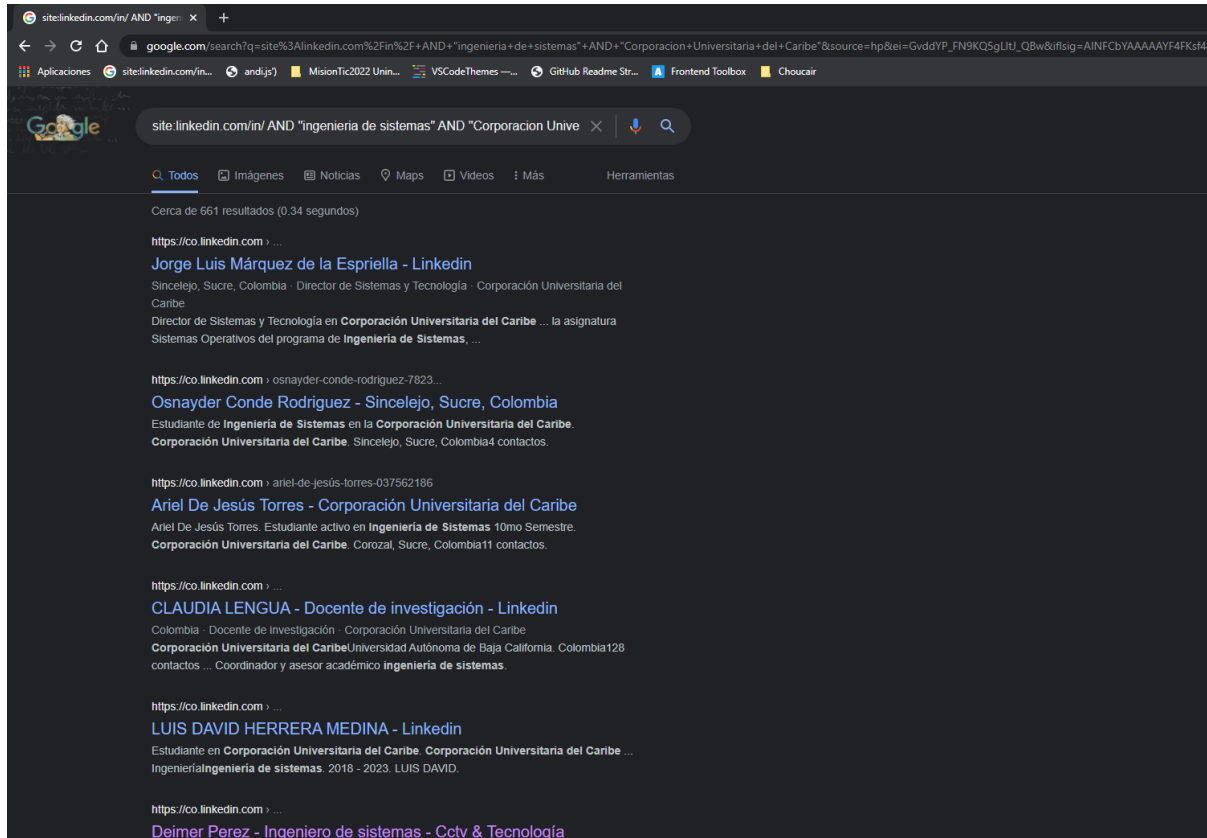
```
# Method to login in linkedin and start to scraper
def login(self):
    try:
        with open('account.txt', 'r') as f:
            line = f.readlines()
            username = line[0]
            password = line[1]

        self._driver.find_element_by_name('session_key').send_keys(username)
        self._driver.find_element_by_name('session_password').send_keys(password)
        sleep(12)
        self._driver.find_element_by_xpath('//*[@type="submit"]').click()
        sleep(3)
        self._driver.get(
            'https://www.google.com/search?q=site%3alinkedin.com%2Fin%2FAND+%22ingenieria+de+sistemas%22+AND+%22Corporacion+Universitaria+del+Caribe%22&source=hp&ei=Gvddy'
        )
    except NoSuchElementException as ex:
        print(ex.msg)
```

Fuente: Elaboración propia

Figura 9.

Navegador en ejecución



Fuente: Elaboración propia

Realización de prueba

Tabla 3.

Ejecucion de pruebas

Pasos de ejecución	<ol style="list-style-type: none"> 1. Ingresar credenciales en la página principal de LinkedIn. 2. Redirección a Google 3. Buscar por medio de palabras clave los perfiles de graduados del programa de ingeniería de sistemas en la Corporación Universitaria del Caribe
Resultado esperado	<ol style="list-style-type: none"> 1. Ingreso de usuario correctamente y visualización del home de LinkedIn. 2. Redireccionamiento al sitio de Google y búsqueda por medio de palabras claves.
Resultado de evaluación de la prueba	Exitosa / OK

Fuente: Elaboración propia.

3.5.1.3. Acceder a cada perfil del resultado de búsqueda.


En esta parte, se busca acceder a los resultados de Google y entrar a cada perfil, donde se valida si el graduado ha estudiado en la CORPORACIÓN UNIVERSITARIA DEL CARIBE y que a la vez cuente con perfil de Ingeniería de Sistemas.

Un ejemplo de los perfiles que son aceptados y scrapeados es el siguiente:

Figura 10.

Sección educación de perfil a scrapear.


Education



Universidad Internacional Iberoamericana (UNINI PR)

Maestría en Dirección Estratégica. Especialidad: Tecnologías de la Información · Tecnología de la información


2013 - 2016



Corporación Universitaria del Caribe

Ingeniero de Sistemas · Computer Engineering

2005 - 2011



Fuente: Elaboración propia

Perfiles no aceptados y no se procede a scrapear, ya que no cumplen con las validaciones anteriormente mencionadas, como haber estudiado en la institución y ser graduado del programa de Ingeniería de Sistemas.

Figura 11.

Perfil que no cumple con los requisitos a scrapear: Ser estudiante de la institución y a la vez haber tomado la carrera de Ingeniería de Sistemas.

Educación



Universidad Autónoma de Baja California
Doctorado gerencia y política educativa, Educación
2016 – 2019



Universidad de Santander
Tecnología educativa, Tecnología de la información
2012 – 2015



Universidad Distrital Francisco José de Caldas
Ingeniería, Ingeniería de software
1999 – 2000



Universidad de Córdoba (CO)
Licenciatura Informática
1996 – 2000



UNIVERSIDAD AMERICANA DE EUROPA (UNAE)
Doctorado en informática
2020

Fuente: Elaboración propia.

Figura 12.

Valida si el perfil cuenta con las validaciones especificadas anteriormente.

```
# Validates if the student's profile studied at cecar
def studied_at_university(self):
    elements_education = len(self._driver.find_elements_by_xpath('//section[@id="education-section"]/ul/li'))
    print('Numero de elementos de universidades de educacion encontrados:', elements_education)
    sleep(2)
    if elements_education == 1:
        university_name = ''
        university_name = (self._driver.find_element_by_xpath('//section[@id="education-section"]/ul/li//h3').text).lower()
        print(f'Universidad: {university_name}')
        sleep(2)
        if self.is_data_exist(university_name, 'static/university.csv', 1):
            return True
        else:
            return False
    elif elements_education > 1:
        university_name = []
        for i in range(elements_education):
            university_name.append((self._driver.find_element_by_xpath(f'//section[@id="education-section"]/ul/li[{i + 1}]/h3').text).lower())
        print(f'Universidades: {university_name}')
        sleep(3)
        if self.is_data_exist(university_name, 'static/university.csv', elements_education):
            return True
        else:
            return False
```

Fuente: Elaboración propia.

Figura 13.

Valida si el perfil cuenta con las validaciones especificadas anteriormente.

```
# Validates if the student's profile is valid if the student studied at cecar and is studying systems engineering.
def studied_career_at_university(self):
    try:
        elements_career = len(self._driver.find_elements_by_xpath('//section//section/ul/li/div/div/a/div[2]/div/p[1]/span[2]'))
        print('Numero de elementos de carreras encontrados:', elements_career)
        sleep(2)
        if elements_career == 1:
            career_degree = (self._driver.find_element_by_xpath('//section[@id="education-section"]/ul/li//div[@class="pv-entity__degree-info"]/p[contains(@class, "pv-enti-')]').text).lower()
            print(f'Carrera: {career_degree}')
            sleep(2)
            # Is passed by parameters career_degree, file of career with a length of 1
            return self.is_data_exist(career_degree, 'static/career.csv', 1)
        else:
            career_degree = [(self._driver.find_element_by_xpath(f'//section//section/ul/li[{i + 1}]/div/div/a/div[2]/div/p[1]/span[2]')).text).lower() for i in range(elements_career)]
            print(f'Carreras: {career_degree}')
            sleep(3)
            return self.is_data_exist(career_degree, 'static/career.csv', elements_career)
    except:
        print('Not found career degree')
```

Fuente: Elaboración propia.

Figura 14.

Valida si el perfil cuenta con las validaciones especificadas anteriormente.

```
# Validate if the student with university education at CECAR
def is_student(self):
    try:
        university = ['corporación universitaria del caribe', 'cecar', 'corporación universitaria del caribe cecar',
                    "corporación universitaria del caribe - cecar", 'corporación universitaria del caribe "cecar"']
        elements_education = len(self._driver.find_elements_by_xpath('//section[@id="education-section"]/ul/li'))
        if elements_education == 1:
            university_name = ''
            university_name = (self._driver.find_element_by_xpath('//section[@id="education-section"]/ul/li//h3').text).lower()
            print(f'Universidad: {university_name}')
            sleep(2)
            if university_name in university:
                return True
            else:
                return False
        elif elements_education > 1:
            count = 0
            university_name = []
            for i in range(elements_education):
                university_name.append((self._driver.find_element_by_xpath(f'//section[@id="education-section"]/ul/li[{i+1}]/h3').text).lower())
            print(f'Universidades: {university_name}')
            sleep(3)
            for i in university_name:
                if i in university:
                    count += 1
            if count > 0:
                return True
            else:
                return False
    except:
        print("Not found university name")
```

Fuente: Elaboración propia.

Figura 15.

Valida si el perfil cuenta con las validaciones especificadas anteriormente.

```
# Validste if student contains a degree as System engineer, engineer or others
def is_student_career(self):
    try:
        career = ['ingeniería de sistemas', 'ingeniería', 'ingeniero', 'ingeniero de sistemas', 'grado de ingeniería',
                  'grado en ingeniería de sistemas', 'grado en ingeniería', 'ciclo formativo de grado superior',
                  'ingeniería de software', 'ingeniero de software', 'diplomatura', 'desarrollo de aplicativos móviles',
                  'grado', 'ingeniera de sistemas', 'ingeniera de sistemas', 'ingeniería de sistemas', 'software engineer',
                  'ingeniera de sistemas (systems engineer)', "engineer's degree", 'ingeniería de software', 'grado en ingeniería']
        elements_career = len(self._driver.find_elements_by_xpath('//section//section/ul/li/div/div/a/div[2]/div/p[1]/span[2]'))
        if elements_career == 1:
            career_degree = self._driver.find_element_by_xpath('//section[id="education-section"]/ul/li//div[@class="pv-entity_degree-info"]/p[contains(@class, "pv-enti
            print(f'Carrera: {career_degree}')
            if career_degree in career:
                return True
            else:
                return False
        else:
            count = 0
            career_degree = [self._driver.find_element_by_xpath(f'//section//section/ul/li[{i+1}]/div/div/a/div[2]/div/p[1]/span[2]').text.lower() for i in range(element
            print(f'Carreras: {career_degree}')
            sleep(2)
            for i in career_degree:
                if i in career:
                    count += 1
            if count > 0:
                return True
            else:
                return False
    except NoSuchElementException as ex:
        print(ex.msg)
```

Fuente: Elaboración propia.

Realización de prueba

Tabla 4.

Ejecucion de pruebas

Pasos de ejecución	<ol style="list-style-type: none"> 1. Entrar al perfil. 2. Validar que haya cursado la carrera de Ingeniería de Sistemas en la Corporación Universitaria del Caribe.
Resultado esperado	<ol style="list-style-type: none"> 1. Perfil con carrera cursada en Ingeniería de Sistemas en la Corporación Universitaria del Caribe
Resultado de evaluación de la prueba	Exitosa / OK

Fuente: Elaboración propia.

3.5.1.4. Validación si la persona ya se encuentra almacenada en el archivo JSON.

En esta parte, se valida si el perfil ya se encuentra almacenado en el archivo JSON, y verifica si cuenta con una actualización en sus diferentes secciones de Experiencia, educación y/o certificados y licencias, para realizar la actualización de esta misma.

Figura 16.

Validación si ya se encuentre en el archivo JSON.

```
# Read a file csv and validate if the data which pass as arg exist at file
# data -> data to validate
# file -> file to read
# length -> length of element found
def is_data_exist(self, data, file, length):
    with open(file, "r+", encoding="utf-8") as f:
        count = 0
        line = f.readline()
        line = line.split(',')
        if length == 1:
            if data in line:
                return True
            else:
                return False
        else:
            for i in data:
                if i in line:
                    count += 1
            if count > 0:
                return True
            else:
                return False
```

Fuente: Elaboración propia.

Realización de prueba

Tabla 5.

Ejecucion de pruebas

Pasos de ejecución	<ol style="list-style-type: none">1. Validar si el graduado se encuentra almacenado en el archivo JSON.2. Verificar si cuenta con un nuevo elemento en sus diferentes secciones del perfil
Resultado esperado	<ol style="list-style-type: none">1. Si el usuario se encuentra almacenado en el archivo JSON y no cuenta con un nuevo elemento, se procede a salir del perfil de LinkedIn y proceder con el siguiente.2. Si se encuentra un nuevo elemento, procede a agregarlo al archivo JSON
Resultado de evaluación de la prueba	Exitosa / OK

Fuente: Elaboración propia.

3.5.1.5. Extracción de datos en la sección de Experiencia.

En esta sección, se procede a realizar la respectiva extracción de datos de los diferentes perfiles que cumplan con las validaciones anteriormente mencionadas.

Figura 17.

Bloque de código el cual extrae la sección de experiencias.

```
def get_data_profile(self):

    list_experience = []
    list_description = []
    list_education = []
    list_certification = []

    print("*** * 100)
    name = self._driver.find_element_by_xpath('//main/div/section/div[2]/div[2]/div/div[1]/h1').text
    career = self._driver.find_element_by_xpath('//main/div/section/div[2]/div[2]/div/div[2]').text
    url_profile = self._driver.current_url
    print(f'Nombre: {name} --- URL: {url_profile}')

    self.exist_button()
    self.exist_button_education()
    self.exist_button_certification()

    elements_experience = len(self._driver.find_elements_by_xpath('//section[@id="experience-section"]/ul/li/section[starts-with(@id, 1) or starts-with(@id, 2) or starts-with(@id, 3)]/ul/li/section[contains(@id, "member")]'))
    elements_experience_extend = len(self._driver.find_elements_by_xpath('//section[@id="experience-section"]/ul/li/section[contains(@id, "member")]'))
    elements_education = len(self._driver.find_elements_by_xpath('//section[@id="education-section"]/ul/li'))
    elements_certifications = len(self._driver.find_elements_by_xpath('//section[@id="certifications-section"]/ul/li'))

    if self.studied_at_university() and self.studied_career_at_university() and self._files.student_exists(name):
        # Experience section
        try:
            if elements_experience == 1:
                experience_position = self._driver.find_element_by_xpath('//section[@id="experience-section"]/ul/li/h3').text
                experience_company = self._driver.find_element_by_xpath('//section[@id="experience-section"]/ul/li/p[contains(@class, "pv-entity__secondary-title t-14")]')
                experience_date = self._driver.find_element_by_xpath('//section[@id="experience-section"]/ul/li/h4[contains(@class, "pv-entity__date-range")]')
                experience = {
                    "responsibility": experience_position,
                    "company": experience_company,
                    "duration": experience_date
                }
                list_experience.append(experience)
            else:
                for i in range(elements_experience):
                    experience_position = self._driver.find_element_by_xpath(f'//section[@id="experience-section"]/ul/li[{i + 1}]/section[starts-with(@id, 1) or starts-with(@id, 2) or starts-with(@id, 3)]/ul/li/section[contains(@id, "member")]')
                    experience_company = self._driver.find_element_by_xpath(f'//section[@id="experience-section"]/ul/li[{i + 1}]/section[starts-with(@id, 1) or starts-with(@id, 2) or starts-with(@id, 3)]/ul/li/section[contains(@id, "member")]')
                    experience_date = self._driver.find_element_by_xpath(f'//section[@id="experience-section"]/ul/li[{i + 1}]/section[starts-with(@id, 1) or starts-with(@id, 2) or starts-with(@id, 3)]/ul/li/section[contains(@id, "member")]')
                    experience = {
                        "responsibility": experience_position,
                        "company": experience_company,
                        "duration": experience_date
                    }
                    list_experience.append(experience)
        except:
            pass
```

Fuente: Elaboración propia.

Figura 18.

Bloque de código que extra la sección de educación.

```
# Section Education
try:
    if elements_education == 1:
        education_name = self._driver.find_element_by_xpath('//section[@id="education-section"]/ul/li//h3').text

        entity_degree_comma = self._driver.find_element_by_xpath('//section[@id="education-section"]/ul/li/div[@class="pv-entity__degree-info"]/p/span[@class="pv-

        education_description = entity_degree_comma

        education_time_from = self._driver.find_element_by_xpath('//section[@id="education-section"]/ul/li/p[contains (@class, "pv-entity__dates")]/span/time[1]')
        education_time_to = self._driver.find_element_by_xpath('//section[@id="education-section"]/ul/li/p[contains (@class, "pv-entity__dates")]/span/time[2]').t
        education_time = f'{education_time_from} - {education_time_to}'

        education = {
            "institution": education_name,
            "degree": education_description,
            "duration": education_time
        }
        list_education.append(education)
    else:
        for i in range(elements_education):
            education_name = self._driver.find_element_by_xpath(f'//section[@id="education-section"]/ul/li[{i + 1}]/h3').text
            #entity_degree_comma = self._driver.find_element_by_xpath(f'//section[@id="education-section"]/ul/li[{i + 1}]/div[@class="pv-entity__degree-info"]/p/s
            entity_degree_comma = self._driver.find_element_by_xpath(f'//section//section/ul/li[{i+1}]/div/div/a/div[2]/div/p[1]/span[2]').text

            education_description = entity_degree_comma

            education_time_from = self._driver.find_element_by_xpath(f'//section[@id="education-section"]/ul/li[{i + 1}]/p[contains (@class, "pv-entity__dates")]/
            education_time_to = self._driver.find_element_by_xpath(f'//section[@id="education-section"]/ul/li[{i + 1}]/p[contains (@class, "pv-entity__dates")]/sp
            education_time = f'{education_time_from} - {education_time_to}'

            education = {
                "institution": education_name,
                "degree": education_description,
                "duration": education_time
            }
            list_education.append(education)
except NoSuchElementException as ex:
    print(ex.msg)
```

Fuente: Elaboración propia.

Figura 19.

Bloque de código que extra la sección de certificaciones

```
# Certifications section
try:
    if elements_certifications == 1:
        name_certification = self_driver.find_element_by_xpath('//section[@id="certifications-section"]/h3').text
        institution_certification = self_driver.find_element_by_xpath('//section[@id="certifications-section"]/p[1]/span[2]').text
        duration_certification = self_driver.find_element_by_xpath('//section[@id="certifications-section"]/p[2]/span[2]').text
        certifications = {
            "certification": name_certification,
            "institution": institution_certification,
            "duration": duration_certification
        }
        list_certification.append(certifications)
    else:
        for i in range(elements_certifications):
            name_certification = self_driver.find_element_by_xpath(f'//section[@id="certifications-section"]/ul/li[{i+1}]/h3').text
            institution_certification = self_driver.find_element_by_xpath(f'//section[@id="certifications-section"]/ul/li[{i+1}]/p[1]/span[2]').text
            duration_certification = self_driver.find_element_by_xpath(f'//section[@id="certifications-section"]/ul/li[{i+1}]/p[2]/span[2]').text
            certifications = {
                "certification": name_certification,
                "institution": institution_certification,
                "duration": duration_certification
            }
            list_certification.append(certifications)
except NoSuchElementException as ex:
    print(ex.msg)
```

Finalmente, se procede a guardar los datos dentro del archivo JSON

Figura 20.

Guardar datos en el archivo JSON.

```
data = {
    "name": name,
    "career": career,
    "url": url_profile,
    "element_experience": elements_experience + elements_experience_extend,
    "elements_education": elements_education,
    "elements_certification": elements_certifications,
    "work": list_experience,
    "education": list_education,
    "certification": list_certification
}
data = json.dumps(data, ensure_ascii=False, indent=4)
self_files.write_file(data)
sleep(19)
```

Fuente: Elaboración propia

Figura 21.

Ejemplo estructura de un bloque de datos almacenados en JSON

```
"name": "DANIEL ERNESTO FRAGOSO AMARIZ",  
"career": "Ingeniero Especialista en la Subdirección Gestión de Redes Sociales e Informalidad del Instituto Para la Economía Social IPES",  
"url": "https://www.linkedin.com/in/daniel-ernesto-fragoso-amariz-bb02a7132/?originalSubdomain=co",  
"element_experience": 9,  
"elements_education": 2,  
"elements_certification": 0,
```

Fuente: Elaboración propia.

Figura 22.

Sección Experiencia.

```
"work": [  
  {  
    "responsibility": "Ingeniero Especialista",  
    "company": "Instituto Para la Economía Social IPES Profesional independiente",  
    "duration": "feb 2020 - actualidad"  
  },  
  {  
    "responsibility": "Ingeniero de Sistemas y Estratega de marketing Digital.",  
    "company": "Consultorias de Sistema y estrategias digitales",  
    "duration": "nov 2018 - actualidad"  
  },  
  {  
    "responsibility": "Ingeniero de Sistemas",  
    "company": "Ministerio del Interior",  
    "duration": "oct 2017 - nov 2018"  
  },  
  {  
    "responsibility": "Docente",  
    "company": "Politecnico Internacional",  
    "duration": "abr 2018 - sept 2018"  
  },  
  {  
    "responsibility": "Ingeniero de Sistemas",  
    "company": "SUPERINTENDENCIA DE PUERTOS Y TRANSPORTES",  
    "duration": "may 2015 - dic 2016"  
  },  
  {  
    "responsibility": "Ingeniero de soporte",  
    "company": "MUNICIPIO HATILLO DE LOBA - BOLÍVAR",  
    "duration": "feb 2013 - oct 2015"  
  }  
]
```

Fuente: Elaboración propia.

Figura 23.

Sección educación.

```
"education": [  
  {  
    "institution": "Universidad Sergio Arboleda",  
    "degree": "Especialista Gerencia de Marketing",  
    "duration": "2016 - 2017"  
  },  
  {  
    "institution": "Corporación Universitaria del Caribe",  
    "degree": "Ingeniero de Sistemas",  
    "duration": "2006 - 2011"  
  }  
],
```

Fuente: Elaboración propia

Figura 24.

Sección certificaciones.

```
"certification": [  
  {  
    "certification": "Curso de Introducción al Desarrollo Web: HTML y CSS",  
    "institution": "Google Activate",  
    "duration": "Expedición: mar 2019Sin fecha de vencimiento"  
  }  
]
```

Fuente: Elaboración propia

Realización de prueba

Tabla 6.

Ejecucion de pruebas

Pasos de ejecución	<ol style="list-style-type: none"> 1. Extracción de datos satisfactoriamente 2. Ingreso de datos en el archivo JSON
Resultado esperado	<ol style="list-style-type: none"> 1. Datos extraídos y almacenados en el archivo JSON
Resultado de evaluación de la prueba	Exitosa / OK

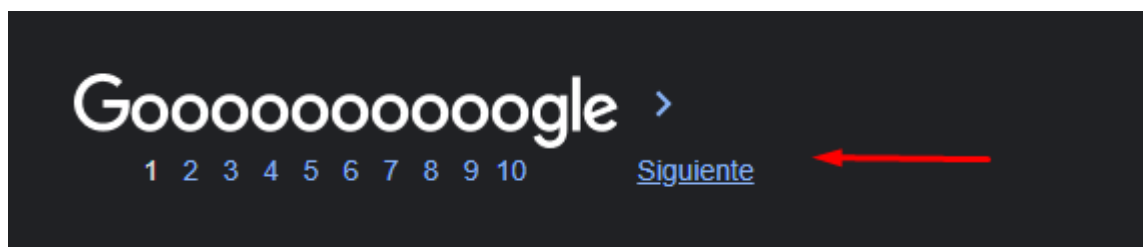
Fuente: Elaboración propia

3.5.1.6. Navegación entre páginas de búsqueda de Google.

En esta sección, realiza las acciones de entrar en las siguientes páginas de búsqueda, con el fin de seguir extrayendo datos de los perfiles de LinkedIn de los graduados del programa de Ingeniería de Sistemas de la Corporación Universitaria Del Caribe.

Figura 25.

Páginas de resultados de Google.



Fuente: Elaboración propia.

Figura 26.

Bloque de código que realiza las acciones de dar clic en las diferentes páginas de resultados de Google.

```
def profile(self):
    url_current = ['https://www.linkedin.com/feed/?trk=people-guest_profile-result-card_result-card_full-click',
                  'https://www.linkedin.com/']
    ]
    get_data = GetDataProfile(self._driver)
    elements_profile = len(self._driver.find_elements_by_xpath(
        '//div[7]/div/div[9]/div[1]/div/div[2]/div[2]/div/div/div/div/div[1]/a'))
    page = 2
    url = ''

    self._files.create_json_file()
    while page < 30:
        get_data.get_link_name_profile()
        sleep(1)
        print("*" * 100)
        print(page)
        print("*" * 100)
        try:
            for profile in range(elements_profile):
                profile_student = self._driver.find_element_by_xpath(
                    f'//div[7]/div/div[9]/div[1]/div/div[2]/div[2]/div/div/div[{profile + 1}]/div/div/div[1]/a')
                self._driver.execute_script("arguments[0].click();", profile_student)
                sleep(6)
                url = self._driver.current_url
                print(url)
                name = len(self._driver.find_elements_by_xpath('//main/div/section/div[2]/div[2]/div/div[1]/h1'))
                print(f'Contador nombre: {name}')
                if url in url_current or name == 0:
                    sleep(3)
                    self._driver.execute_script("window.history.go(-1)")
                else:
                    sleep(6)
                    get_data.get_data_profile()
                    self._driver.execute_script("window.history.go(-1)")
                    sleep(2)
                navigator_page = self._driver.find_element_by_link_text(f'{page}')
                self._driver.execute_script("arguments[0].click();", navigator_page)
                page += 1
        except NoSuchElementException as ex:
            print(ex.msg)
```

Fuente: Elaboración propia.

Tabla 7.

Ejecucion de pruebas

Pasos de ejecución	<ol style="list-style-type: none"> 1. Da clic en la siguiente página de búsqueda de Google. 2. Carga la siguiente página de búsqueda de Google
Resultado esperado	<ol style="list-style-type: none"> 1. Realiza los pasos correctamente hasta llegar a la siguiente página de búsqueda de Google y sigue con la respectiva extracción de datos.
Resultado de evaluación de la prueba	Exitosa / OK

Fuente: Elaboración propia.

3.5.1.7. Búsqueda por medio de diferentes palabras claves.

Como bien se sabe, Google cuenta con un total de páginas de resultados las cuales oscilan entre 27 a 30 páginas. Es por esto que se optó por realizar diferentes búsquedas con diferentes palabras claves para obtener la mayor cantidad de perfiles Scrapeados, a medida que se cumpliera el ciclo de ejecución, para así contar con más datos para su respectivo análisis.

Figura 27.

Bloque de código de búsqueda por medio de diferentes palabras claves

```

self_driver.get(
    "https://www.google.com/search?q=site%3Alinkedin.com%2Ffin%2F+MD+K22sistemas%22+MD+K22CorporacionUniversitaria+de1+Caribe%22&rl=evy2YL6Y696CwbKp-1mURAK8oq=site%3
page = 2
elements_profile = len(self_driver.find_elements_by_xpath(
    '//div[7]/div/div[9]/div[11]/div/div[2]/div[2]/div/div/div/div[11]/e')
while page < 30:
    print("*** + 100)
    print(page)
    print("*** + 100)
    get_data.get_link_name_profile()
    sleep(2)
    try:
        for profile in range(elements_profile):
            profile_student = self_driver.find_element_by_xpath(
                f'//div[7]/div/div[9]/div[11]/div/div[2]/div[2]/div/div/div[{profile + 1}]/div/div/div[11]/e')
            self_driver.execute_script("arguments[0].click()", profile_student)
            sleep(6)
            url = self_driver.current_url
            print(url)
            name = len(self_driver.find_elements_by_xpath('//*[@main/div/div/div[2]/div[2]/div/div[11]/h1']))
            print(f'Contador nombre: {name}')
            if url != url_current or name == 0:
                sleep(1)
                self_driver.execute_script("window.history.go(-1)")
            else:
                sleep(6)
                get_data.get_data_profile()
                self_driver.execute_script("window.history.go(-1)")
                sleep(2)
            navigator_page = self_driver.find_element_by_link_text(f'{page}')
            self_driver.execute_script("arguments[0].click()", navigator_page)
            page += 1
    except NoSuchElementException as ex:
        print(ex.msg)
    
```

Fuente: Elaboración propia

Figura 28.

Bloque de código de búsqueda por medio de diferentes palabras claves.

```
self._driver.get(
    'https://www.google.com/search?q=site:linkedin.com/in/*AND*2desarrollador*2+AND*22corporacion+universitaria+del+Caribe*22&ei=7o29YPTPI-Qo5NoPhFe3gAs&start=0&sa
page = 2
elements_profile = len(self._driver.find_elements_by_xpath(
    '//div[7]/div/div[9]/div[1]/div/div[2]/div[2]/div/div/div/div[1]/a'))
while page < 30:
    print("*** + 100)
    print(page)
    print("*** + 100)
    get_data.get_link_name_profile()
    sleep(2)
    try:
        for profile in range(elements_profile):
            profile_student = self._driver.find_element_by_xpath(
                f'//div[7]/div/div[9]/div[1]/div/div[2]/div[2]/div/div/div[{profile + 1}]/div/div/div[1]/a')
            self._driver.execute_script("arguments[0].click();", profile_student)
            sleep(6)
            url = self._driver.current_url
            print(url)
            name = len(self._driver.find_elements_by_xpath('//main/div/section/div[2]/div[2]/div/div[1]/h1'))
            print(f'Contador nombre: {name}')
            if url in url_current or name == 0:
                sleep(2)
                self._driver.execute_script("window.history.go(-1)")
            else:
                sleep(5)
                get_data.get_data_profile()
                self._driver.execute_script("window.history.go(-1)")
                sleep(2)
            navigator_page = self._driver.find_element_by_link_text(f'{page}')
            self._driver.execute_script("arguments[0].click();", navigator_page)
            page += 1
    except NoSuchElementException as ex:
        print(ex.msg)
```

Fuente: Elaboración propia

Figura 29.

Bloque de código de búsqueda por medio de diferentes palabras claves.

```
self._driver.get(
    'https://www.google.com/search?q=site:31linkedin.com*2Fin&ZF+AND*22Ingenieria+de+sistemas*22+AND*22CECAR*22&biw=1440&bih=447&ei=ayAYOGu6o5NoP6Zuj-AE&og=site&sa
page = 2
elements_profile = len(self._driver.find_elements_by_xpath(
    '//div[7]/div/div[9]/div[1]/div/div[2]/div[2]/div/div/div/div[1]/a'))
while page < 13:
    print("*** + 100)
    print(page)
    print("*** + 100)
    get_data.get_link_name_profile()
    sleep(2)
    try:
        for profile in range(elements_profile):
            profile_student = self._driver.find_element_by_xpath(
                f'//div[7]/div/div[9]/div[1]/div/div[2]/div[2]/div/div/div[{profile + 1}]/div/div/div[1]/a')
            self._driver.execute_script("arguments[0].click();", profile_student)
            sleep(5)
            url = self._driver.current_url
            print(url)
            name = len(self._driver.find_elements_by_xpath('//main/div/section/div[2]/div[2]/div/div[1]/h1'))
            print(f'Contador nombre: {name}')
            if url in url_current or name == 0:
                sleep(2)
                self._driver.execute_script("window.history.go(-1)")
            else:
                sleep(6)
                get_data.get_data_profile()
                self._driver.execute_script("window.history.go(-1)")
                sleep(4)
            navigator_page = self._driver.find_element_by_link_text(f'{page}')
            self._driver.execute_script("arguments[0].click();", navigator_page)
            page += 1
    except NoSuchElementException as ex:
        print(ex.msg)
```

Fuente: Elaboración propia

Tabla 8.

Ejecucion de pruebas

Pasos de ejecución	<ol style="list-style-type: none"> 1. Cargar diferentes búsquedas con ayuda de diferentes palabras claves. 2. Cargar los resultados.
Resultado esperado	<ol style="list-style-type: none"> 1. Cargar resultados correctamente.
Resultado de evaluación de la prueba	Exitosa / OK

Fuente: Elaboración propia.

3.5.1.8. Creación de archivo JSON.

En esta sección se realizó el proceso de creación del archivo Json el cual contendrá todos los datos extraídos de cada perfil de los graduados.

Figura 30.

Bloque de código que crea, elimina, actualiza el archivo JSON.

```

import os

class Files:

    # Write data in json file
    def write_file(self, data):
        with open("data.json", "a+", encoding="utf-8") as f:
            f.write(f"{data},")

    # Delete json file when its create before to run the project
    def delete_json(self):
        if os.path.exists("data.json"):
            return os.remove("data.json")

    # Method to create json file that start with "[" for it take format of json
    def create_json_file(self):
        with open("data.json", "a+", encoding="utf-8") as f:
            f.write("[\n")
            f.close()

    # Open data.json and read, then validate if found
    def close_json_file(self):
        # If not exist file, show error and pass to next line, when read data.json file and then write in new file
        # called new_data.json
        try:
            os.remove("new_data.json")
        except FileNotFoundError as fn:
            print(fn)

        with open("data.json", "r+", encoding="utf-8") as input:
            with open("new_data.json", "a+", encoding="utf-8") as f:
                for line in input:
                    if line != "],":
                        f.write(line)
                    f.write("],")

    def student_exists(self, name):
        count = 0
        with open("data.json", "r+", encoding="utf-8") as f:
            for line in f:
                if name in line:
                    count += 1
            if count == 0:
                return True
            else:
                return False
    
```

Fuente: Elaboración propia

Realización de pruebas

Tabla 9.

Ejecucion de pruebas

Pasos de ejecución	<ol style="list-style-type: none"> 1. Crear archivo JSON. 2. Escritura de los datos extraídos hacia el archivo JSON con su respectiva estructura
Resultado esperado	<ol style="list-style-type: none"> 1. Escritura de datos con su respectiva estructura
Resultado de evaluación de la prueba	Exitosa / OK

Fuente: Elaboración propia.

3.5.1.9. Clasificación de los datos

En este último proceso, se llevó a cabo la clasificación de los datos obtenidos por medio del web scraping que se realizó en el sitio web LinkedIn por medio de los perfiles de los graduados del programa de Ingeniería de Sistemas.

Según el Proyecto Educativo del Programa (PEP) del programa de Ingeniería de Sistemas de la Corporación Universitaria del Caribe, cuenta con diferentes perfiles de ingreso, profesional y ocupacional o perfiles del aspirante y del egresado. Donde cuenta con nueve (9) perfiles ocupacionales “los cuales desempeñará profesionalmente en organizaciones de los sectores público y privado, de índole regional o nacional.”. (Proyecto Educativo del Programa, 2018)

Estos perfiles son:

- Ingeniero de Desarrollo y Análisis de Software.
- Administrador de Bases de Datos
- Administrador Redes de Computadores
- Ingeniero de Soporte y/o Mantenimiento.
- Administrador de Servicios Informáticos.
- Desarrollador de Soluciones Integrales.

- Desarrollador de Sistemas Informáticos.
- Investigador.
- Gestor de Proyectos de Ingeniería.

Es por esto que se realizó una caracterización de todos los datos y se agrupó por los diferentes perfiles ocupacionales que brinda la Institución, donde por medio de diferentes palabras claves se realizaba esta clasificación de datos, con ayuda de la librería de Python llamada Pandas, el cual hacia este proceso más fácil y eficiente.

Figura 31.

Bloque de código de clasificación de datos scrapeados.

```
def fact_ident_experience(filecsv):
    """
    Extract all data experiences of json, then process it and finally write the clean data to csv file
    """

    filename = "../dataprueba.json"

    header = ["responsibility", "duration", "location"]

    # Extract all data of key responsibility
    with open(filename, 'rb') as f:
        data_json_responsibility = ijson.items(f, 'item.work.item.responsibility')
        responsibility = [obj for obj in data_json_responsibility]

    # Extract all data of key responsibility but, extend
    with open(filename, 'rb') as f:
        data_json_responsibility_extend = ijson.items(f, 'item.work.item.description.item.responsibility')
        responsibility_extend = [obj for obj in data_json_responsibility_extend]

    # concat list of responsibility and responsibility_extend
    result_responsibility = responsibility + responsibility_extend

    # Extract all data from key duration in key work
    with open(filename, 'rb') as f:
        data_json_duration = ijson.items(f, 'item.work.item.duration')
        duration = [obj for obj in data_json_duration]

    # Extract all data from key duration extended in key work
    with open(filename, 'rb') as f:
        data_json_duration_extend = ijson.items(f, 'item.work.item.description.item.duration')
        duration_extend = [obj for obj in data_json_duration_extend]

    # concat list duration with duration extended
    result_duration = duration + duration_extend

    #
    with open(filename, 'rb') as f:
        data_json_location = ijson.items(f, 'item.work.item.description.item.location')
        location = [obj for obj in data_json_location]

    d = [result_responsibility, result_duration, location]

    export_data = zip_longest(*d, fillvalue='Null')
```

Fuente: Elaboración propia.

Figura 32.

Bloque de código de clasificación de datos scrapeados.

```
def fact_ident_education(filecsv):
    """Extract all data education of json, then process it and finally write the clean data to csv file called
    data_process_education.csv"""

    header = ['institution', 'degree', 'duration']

    with open(FILE_JSON_DATA, 'rb') as f:
        data_json_institution = ijson.items(f, 'item.education.item.institution')
        institution = [obj for obj in data_json_institution]

    with open(FILE_JSON_DATA, 'rb') as f:
        data_json_degree = ijson.items(f, 'item.education.item.degree')
        degree = [obj for obj in data_json_degree]

    with open(FILE_JSON_DATA, 'rb') as f:
        data_json_duration = ijson.items(f, 'item.education.item.duration')
        duration = [obj for obj in data_json_duration]

    d = [institution, degree, duration]

    export_data = zip_longest(*d, fillvalue='Null')

    with open(filecsv, 'w+', encoding='utf-8') as csvfile:
        writer = csv.writer(csvfile)
        writer.writerow(header)
        writer.writerows(export_data)
```

Fuente: Elaboración propia.

Figura 33.

Bloque de código de clasificación de datos scrapeados.

```
def fact_ident_certifications(filecsv):
    """Extract all data certifications of json, then process it and finally write the clean data to csv file called
    data_process_certification.csv"""

    header = ['certification', 'institution', 'duration']

    with open(FILE_JSON_DATA, 'rb') as f:
        data_json_certification = ijson.items(f, 'item.certification.item.certification')
        certification = [obj for obj in data_json_certification]

    with open(FILE_JSON_DATA, 'rb') as f:
        data_json_institution = ijson.items(f, 'item.certification.item.institution')
        institution = [obj for obj in data_json_institution]

    with open(FILE_JSON_DATA, 'rb') as f:
        data_json_duration = ijson.items(f, 'item.certification.item.duration')
        duration = [obj for obj in data_json_duration]

    d = [certification, institution, duration]

    export_data = zip_longest(*d, fillvalue='Null')

    with open(filecsv, 'w+', encoding='utf-8') as csvfile:
        writer = csv.writer(csvfile)
        writer.writerow(header)
        writer.writerows(export_data)
```

Fuente: Elaboración propia.

Figura 34.

Bloque de código de clasificación de datos scrapeados.

```
def data_classification_experience():
    desarrollador, admin_bd, admin_red, soporte, admin_servicio, dev_soluciones, dev_sistemas, investigador, gestor_proyec, others = [], [], [], [], [], [], [], [], []
    count_desarrollador, count_admin_bd, count_admin_red, count_soporte, count_adm_servicio, count_dev_solucion, count_dev_sistemas, count_investigador, count_gest_proyec, countador_total = 0
    with open(FILE_CSV_EXPERIENCE, "r", encoding="utf-8") as file_csv:
        reader = csv.DictReader(file_csv)
        for row in reader:
            row_res = row["responsibility"].lower().split(" ")

            if row_res.__contains__("null"):
                continue

            # Ingeniero de Desarrollo y Análisis de Software.
            if row_res[0] in LIST_ING_DESARROLLO_ANALISIS or row_res.__contains__("consultor") or row_res.__contains__("developer") or row_res.__contains__("analyst") or row_r
                count_desarrollador += 1
                desarrollador.append(row["responsibility"])
            # Administrador de Bases de datos
            elif row_res.__contains__("dato") or row_res.__contains__("datos") or row_res.__contains__("bases"):
                count_admin_bd += 1
                admin_bd.append(row["responsibility"])
            # Administrador redes de computadores
            elif row_res.__contains__("redes") or row_res.__contains__("red") or row_res.__contains__("sysadmin") or row_res.__contains__("systems"):
                count_admin_red += 1
                admin_red.append(row["responsibility"])
            # Ingeniero de Soporte y/o mantenimiento
            elif row_res.__contains__("soporte") or row_res.__contains__("tecnico") or row_res.__contains__("técnico") and not row_res.__contains__("desarrollador") and not ro
                count_soporte += 1
                soporte.append(row["responsibility"])
            # Administrador de servicios informáticos
            elif row_res.__contains__("administrador") or row_res.__contains__("coordinador") or row_res.__contains__("seguridad") or row_res.__contains__("security") or row_r
                count_adm_servicio += 1
                admin_servicio.append(row["responsibility"])
            # Desarrollador de Soluciones Integrales
            elif row_res.__contains__("soluciones") or row_res.__contains__("arquitecto") or row_res.__contains__("innovación"):
                count_dev_solucion += 1
                dev_soluciones.append(row["responsibility"])
            # Desarrollador de Sistemas Informáticos
            elif row_res.__contains__("informatico") or row_res.__contains__("informatica") or row_res.__contains__("sistemas") or row_res.__contains__("webmaster") or row_res
                count_dev_sistemas += 1
                dev_sistemas.append(row["responsibility"])
            # Investigador
            elif row_res.__contains__("docente") or row_res.__contains__("investigador") or row_res.__contains__("docencia"):
                count_investigador += 1
                investigador.append(row["responsibility"])
```

Fuente: Elaboración propia.

Figura 35.

Bloque de código de clasificación de datos scrapeados.

```
work_profile = desarrollador + admin_bd + admin_red + soporte + admin_servicio + dev_soluciones + dev_sistemas + investigador + gestor_proyec

# Write all work profile that extract from data_process_experience.csv
with open('identity_factor.csv', 'w', encoding='utf-8') as f:
    header = ["responsibility"]
    writer = csv.writer(f)
    d = [work_profile]
    export_data = zip_longest(*d)
    writer.writerow(header)
    writer.writerows(export_data)

# Write others professional profiles that doesnt belong in the university
with open('otros.csv', 'w', encoding='utf-8') as f:
    header = ["responsibility"]
    writer = csv.writer(f)
    d = [others]
    export_data = zip_longest(*d)
    writer.writerow(header)
    writer.writerows(export_data)

return count_desarrollador, count_admin_bd, count_admin_red, count_soporte, count_adm_servicio, count_dev_solucion, count_dev_sistemas, count_investigador, count_gest_proy
```

Fuente: Elaboración propia.

Figura 36.

Bloque de código de clasificación de datos scrapeados.

```
def data_classification_certification():
    desarrollador, admin_bd, admin_red, soporte, admin_servicio, dev_soluciones, dev_sistemas, investigador, gestor_proyec, others = [], [], [], [], [], [], [], []
    count_desarrollador, count_admin_bd, count_admin_red, count_soporte, count_admin_servicio, count_dev_solucion, count_dev_sistemas, count_investigador, count_gest_proyec, co
    contador_total = 0
    write_csv = []

    with open(FILE_CSV_CERTIFICATION, "r", encoding="utf-8") as file_csv:
        reader = csv.DictReader(file_csv)
        for row in reader:
            write_csv.append(row["certification"].lower())
            row_res = row["certification"].lower().split(" ")

            # Ingeniero de Desarrollo y Análisis de Software.
            if row_res.__contains__("progra") or row_res.__contains__("git") or row_res.__contains__("github") or row_res.__contains__("programming") or row_res.__contains__(
                count_desarrollador += 1
                desarrollador.append(row["certification"])

            # Administrador de Bases de datos
            elif row_res.__contains__("dato") or row_res.__contains__("datos") or row_res.__contains__("bases") or row_res.__contains__("sql") or row_res.__contains__("postgre
                count_admin_bd += 1
                admin_bd.append(row["certification"])

            # Administrador redes de computadores
            elif row_res.__contains__("redes") or row_res.__contains__("networking") or row_res.__contains__("support") or row_res.__contains__("cloud") or row_res.__contains
                count_admin_red += 1
                admin_red.append(row["certification"])

            # Ingeniero de Soporte y/o mantenimiento
            elif row_res.__contains__("soporte") or row_res.__contains__("tecnico") or row_res.__contains__("técnico") and not row_res.__contains__("desarrollador") and not ro
                count_soporte += 1
                soporte.append(row["certification"])

            # Administrador de servicios informáticos
            elif row_res.__contains__("administrador") or row_res.__contains__("devops") or row_res.__contains__("ciberseguridad") or row_res.__contains__("coordinador") or ro
                count_admin_servicio += 1
                admin_servicio.append(row["certification"])

            # Desarrollador de Soluciones Integrales
            elif row_res.__contains__("soluciones") or row_res.__contains__("data") or row_res.__contains__("arquitecto") or row_res.__contains__("innovación") or row_res.__co
                count_dev_solucion += 1
                dev_soluciones.append(row["certification"])

            # Desarrollador de Sistemas Informáticos
            elif row_res.__contains__("informatico") or row_res.__contains__("wordpress") or row_res.__contains__("node.js") or row_res.__contains__("nokia") or row_res.__cont
                count_dev_sistemas += 1
                dev_sistemas.append(row["certification"])

            # Investigador
            elif row_res.__contains__("investigador"):
                count_investigador += 1
                investigador.append(row["certification"])

            # Gestor de proyectos de ingeniería
```

Fuente: Elaboración propia.

Figura 37.

Bloque de código de clasificación de datos scrapeados.

```
# Investigador
elif row_res.__contains__("investigador"):
    count_investigador += 1
    investigador.append(row["certification"])

# Gestor de proyectos de ingeniería
elif row_res.__contains__('proyecto') or row_res.__contains__('gestor') or row_res.__contains__('proyectos'):
    count_gest_proyec += 1
    gestor_proyec.append(row["certification"])

else:
    count_others += 1
    others.append(row["certification"])

contador_total += 1 # Count line of csv file
```

Fuente: Elaboración propia.

Figura 38.

Bloque de código de clasificación de datos scrapeados.

```
def calculate_percentage():
    percentage_experience, percentage_certification = [], []
    desarrollador, admin_bd, admin_red, soporte, adm_servicio, dev_solucion, dev_sistemas, investigador, gest_project, others, contador_total = data_classification_experience(
    c_desarrollador, c_admin_bd, c_admin_red, c_soporte, c_adm_servicio, c_dev_solucion, c_dev_sistemas, c_investigador, c_gestor_project, c_others, c_contador_total = data_cl

    # Calculate percentage of all professional profiles
    # % = (cantidad / total) * 100s

    # Percentage experience section
    print("""
    #####
    EXPERIENCE
    #####
    """)

    total = round((desarrollador / contador_total)*100, 2)
    percentage_experience.append(total)

    total = round((admin_bd / contador_total)*100, 2)
    percentage_experience.append(total)

    total = round((admin_red / contador_total)*100, 2)
    percentage_experience.append(total)

    total = round((soporte / contador_total)*100, 2)
    percentage_experience.append(total)

    total = round((adm_servicio / contador_total)*100, 2)
    percentage_experience.append(total)

    total = round((dev_solucion / contador_total)*100, 2)
    percentage_experience.append(total)

    total = round((dev_sistemas / contador_total)*100, 2)
    percentage_experience.append(total)

    total = round((investigador / contador_total)*100, 2)
    percentage_experience.append(total)

    total = round((gest_project / contador_total)*100, 2)
    percentage_experience.append(total)

    total = round((others / contador_total)*100, 2)
```

Fuente: Elaboración propia.

Figura 39.

Bloque de código de clasificación de datos scrapeados.

```
total = round((dev_sistemas / contador_total)*100, 2)
percentage_experience.append(total)

total = round((investigador / contador_total)*100, 2)
percentage_experience.append(total)

total = round((gest_proyect / contador_total)*100, 2)
percentage_experience.append(total)

total = round((others / contador_total)*100, 2)
percentage_experience.append(total)

df = pd.DataFrame({'Profiles':['Ingeniero de Desarrollo y Análisis de Software',
                              'Administrador de Bases de datos',
                              'Administrador redes de computadores',
                              'Ingeniero de Soporte y/o mantenimiento',
                              'Administrador de servicios informáticos',
                              'Desarrollador de Soluciones Integrales',
                              'Desarrollador de Sistemas Informáticos',
                              'Investigador',
                              'Gestor de proyectos de ingeniería',
                              'Otros perfiles'
                              ],
                  'Percentage':[
                              percentage_experience[0],
                              percentage_experience[1],
                              percentage_experience[2],
                              percentage_experience[3],
                              percentage_experience[4],
                              percentage_experience[5],
                              percentage_experience[6],
                              percentage_experience[7],
                              percentage_experience[8],
                              percentage_experience[9],
                              ]
                  })
df.to_excel('experience.xlsx', engine='xlsxwriter')
```

Fuente: Elaboración propia.

Figura 40.

Bloque de código de clasificación de datos scrapeados.

```
print("""
#####
                CERTIFICATIONS
#####
""")

c_total = round((c_desarrollador / c_contador_total)*100, 2)
percentage_certification.append(c_total)

c_total = round((c_admin_bd / c_contador_total)*100, 2)
percentage_certification.append(c_total)

c_total = round((c_admin_red / c_contador_total)*100, 2)
percentage_certification.append(c_total)

c_total = round((c_soporte / c_contador_total)*100, 2)
percentage_certification.append(c_total)

c_total = round((c_adm_servicio / c_contador_total)*100, 2)
percentage_certification.append(c_total)

c_total = round((c_dev_solucion / c_contador_total)*100, 2)
percentage_certification.append(c_total)

c_total = round((c_dev_sistemas / c_contador_total)*100, 2)
percentage_certification.append(c_total)

c_total = round((c_investigador / c_contador_total)*100, 2)
percentage_certification.append(c_total)

c_total = round((c_gestor_proyect / c_contador_total)*100, 2)
percentage_certification.append(c_total)

c_total = round((c_others / c_contador_total)*100, 2)
percentage_certification.append(c_total)
```

Fuente: Elaboración propia.

Figura 41.

Bloque de código de clasificación de datos scrapeados.

```
df = pd.DataFrame({'Profiles':['Ingeniero de Desarrollo y Análisis de Software',
                              'Administrador de Bases de datos',
                              'Administrador redes de computadores',
                              'Ingeniero de Soporte y/o mantenimiento',
                              'Administrador de servicios informáticos',
                              'Desarrollador de Soluciones Integrales',
                              'Desarrollador de Sistemas Informáticos',
                              'Investigador',
                              'Gestor de proyectos de ingeniería',
                              'Otros perfiles'
                              ],
                  'Percentage':[
                              percentage_certification[0],
                              percentage_certification[1],
                              percentage_certification[2],
                              percentage_certification[3],
                              percentage_certification[4],
                              percentage_certification[5],
                              percentage_certification[6],
                              percentage_certification[7],
                              percentage_certification[8],
                              percentage_certification[9],
                              ]
                  })
df.to_excel('certifications.xlsx', engine='xlsxwriter')
#print(df)

print("""
#####
WORKS LOCATION
#####
""")
```

Fuente: Elaboración propia.

Resultados prueba

Tabla 10.

Ejecucion de pruebas

Pasos de ejecución	<ol style="list-style-type: none"> 1. Leer archivo JSON y cargar todos los datos. 2. Procesar los datos y clasificarlos por grupo, asignando su respectivo perfil ocupacional. 3. Ejecutar porcentaje por perfil ocupacional
Resultado esperado	<ol style="list-style-type: none"> 1. Resultado en porcentaje por perfil ocupacional, por las diferentes secciones: experiencia, educación y certificaciones.
Resultado de evaluación de la prueba	Exitosa / OK

Fuente: Elaboración propia.

4. Resultados

Partiendo del primer objetivo del proyecto el cual fue, identificar los requerimientos necesarios para el desarrollo de la herramienta, el cual se obtuvo a través de la entrevista realizada a la Coordinadora María Angelica García Medina la cual contaba con este cargo durante el tiempo que se realizó dicha entrevista, se pudo detallar el proceso y técnicas con los que cuentan hasta el día de hoy la Institución para recolectar los datos de los graduados para así llevar el respectivo seguimiento de su evolución académica y laboral.

Después de haber identificado los requerimientos y datos obtenidos por cada graduado, se procedió a analizar diferentes métodos y tecnologías que ayudarán a construir esta herramienta desde la plataforma de LinkedIn. Para este objetivo en específico, se hizo énfasis y se tuvo diferentes criterios para cada tecnología a utilizar. Estos criterios fueron desde, la versión más estable de cada tecnología como el Lenguaje de programación Python y su librería Pandas específicamente para la etapa de procesamiento de datos y así efectuar una caracterización de los graduados del programa de Ingeniería de Sistemas. De igual manera la herramienta que nos facilitó el proceso de web Scraping la cual fue Selenium ya que Python y esta herramienta se complementan de manera correcta y de igual forma cuentan con una fácil curva de aprendizaje. De igual forma, en el proceso de almacenamiento se utilizaron archivos JSON y la base de datos no relacional Firebase dado que contaban con las mismas características que Python y Selenium de poseer una fácil curva de aprendizaje y hacer el proceso mucho más fácil.

Luego de tener claro y estructurado el diseño de la herramienta se procedió a la construcción de esta. Estos resultados son reflejados más adelante con cada gráfica y tabla que se pudo obtener de todo este proceso de extracción, almacenamiento y procesamiento de datos de los graduados del programa de Ingeniería de Sistemas. Paralelo a esto, se iba realizando el objetivo de ejecutar pruebas unitarias que permitieran asegurar el buen funcionamiento de la herramienta para así suministrar datos aplicables para su posterior análisis con las herramientas ya seleccionadas en el proceso de análisis. Estos resultados también se pueden consultar en el apartado de los anexos, donde se encuentra el código fuente de la herramienta y de igual forma las pruebas unitarias implementadas.

Luego de haber realizado todo el proceso de extracción, procesamiento y almacenamiento de datos, se obtuvieron datos de gran importancia, como son: experiencia laboral que ha tenido el graduado, educación y licencias y certificaciones que ha realizado en toda su evolución educativa como profesional. Como muestra de este ejercicio a continuación se mostrarán datos reales extraídos y que son de gran importancia para la toma de decisiones.

Es importante tener en cuenta los perfiles ocupacionales que brinda la institución al momento de finalizar el proceso educativo en el programa de Ingeniería de sistema, donde estos anteriormente se detallaron en la sección de **Clasificación de los datos**. Estos perfiles ayudan a tener una mayor visión de cómo clasificar y agrupar estos datos ya que cada perfil cuenta con diferentes funciones y tareas que se deben ejecutar en el campo laboral.

Pero antes que todo, se tuvo que realizar diferentes investigaciones de todos estos perfiles ocupacionales, por lo que cada uno de estos se centran en diferentes campos de la Ingeniería de Sistema, para así poder tener mayor claridad y realizar la caracterización de forma correcta agrupando los diferentes datos con respecto al campo que se aplica cada una de estas.

Gracias a los datos obtenidos por medio de los perfiles de LinkedIn de los graduados, se pudo analizar diferentes puntos que ayudan a llevar el seguimiento de los graduados, desde el promedio en que tienden a graduarse los estudiantes del programa de Ingeniería de Sistemas, a que tienden a certificarse, que perfiles ocupacionales se inclinan o les interesa en su evolución educativa como laboral y/o profesional, hasta en qué países o ciudad tienden a desarrollarse como profesional.

Estos resultados fueron obtenidos con ayuda de la herramienta Power BI, la cual facilitó la carga de los datos para su posterior análisis y creación de reporte por medio de diferentes gráficas como de barras, torta o pastel y de igual forma ayudó a graficar de forma geográfica en las locaciones que los graduados han realizado sus experiencias laborales y crecimiento profesional.

4.1. Tiempo que tardan los Estudiantes del programa de Ingeniería de Sistemas en graduarse

La tabla 11, muestra los resultados obtenidos de la extracción de datos realizada a los graduados del programa de Ingeniería de Sistema, donde se evidencia un rango de años que transcurren para poder obtener el título profesional, y de igual manera, cuenta con el recuento de graduados que obtienen este título en los diferentes años.

Tabla 11.

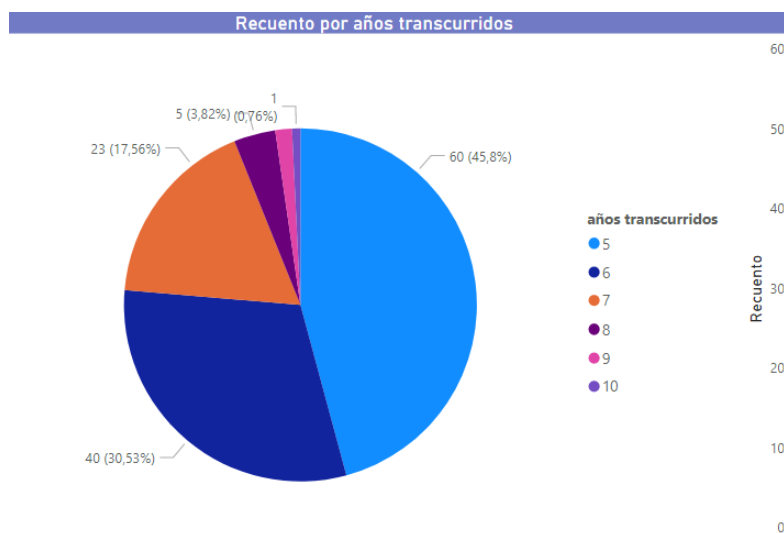
Tabla de resultados del Tiempo que tardan los estudiantes del programa de Ingeniería de sistemas en graduarse

Años transcurridos	Recuento
5	60
6	40
7	23
8	5
9	2
10	1

Fuente: Elaboración propia.

Figura 42.

Gráfico de torta de recuento por años en que se gradúan los estudiantes del programa de Ingeniería de Sistemas.



Fuente: Elaboración propia.

En la figura 42, se puede observar que el 45,8% de los estudiantes del programa de Ingeniería de Sistemas tardan 5 años en graduarse. Adicionalmente la tabla 11 y la figura 42, revelan que el siguiente dato detalla que el 30,53% de los estudiantes tardan 6 años en graduarse. Esto refleja que los estudiantes de ingeniería de sistemas en su mayoría cumplen con los tiempos estipulados por la institución para culminar con los estudios del plan académico.

4.2. Perfiles que tienden a certificarse los graduados del programa de Ingeniería de Sistemas

En la tabla 12 se visualizan los diferentes perfiles profesionales y ocupacionales establecidos por la PEP del programa de Ingeniería de Sistema de la Institución y de igual forma el porcentaje por cada perfil que los graduados tienden a certificarse.

Tabla 12.

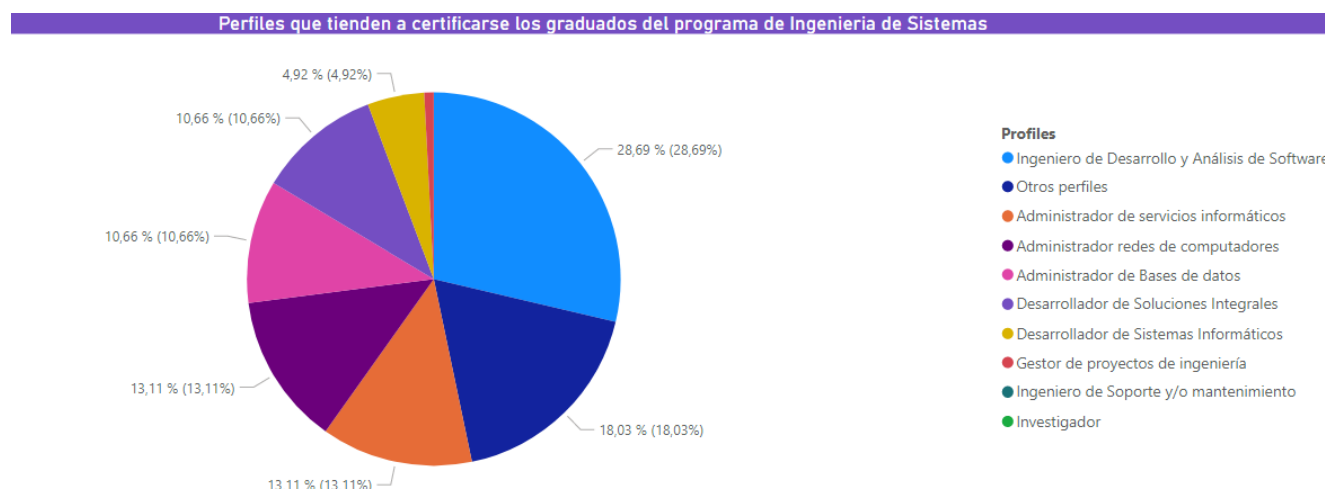
Tabla de resultados de los perfiles que tienden a certificarse los graduados del programa de Ingeniería de sistemas.

Perfiles	Porcentaje
Ingeniero de Desarrollo y Análisis de Software	28.69%
Administrador de Bases de datos	10.66%
Administrador de Redes de Computadora	13.11%
Ingeniero de Soporte y/o Mantenimiento	0.00%
Administrador de Servicios Informáticos	13.11%
Desarrollo de soluciones Integrales	10.66%
Desarrollador de sistemas informáticos	4.92%
Investigador	0.00%
Gestor de proyectos de Ingeniería	0.82%
Otros perfiles	18.03%

Fuente: Elaboración propia

Figura 43.

Gráfico de torta de los perfiles que tienden a certificarse los graduados del programa de Ingeniería de Sistemas.



Fuente: Elaboración propia

En la figura 43, encontramos que el porcentaje mayor de perfiles desarrollados por Ingenieros de sistemas egresados de la institución va acorde al lineamiento del plan académico que presenta la institución la cual es enfocada al desarrollo de software. También es evidente que los perfiles con mayores porcentajes están ligados a las áreas que más énfasis tienen dentro del plan académico.

4.3. Perfiles que tienden a laborar los graduados del programa de Ingeniería de Sistemas

En la tabla 13 se muestra los diferentes perfiles ocupacionales que los graduados del programa de Ingeniería de Sistema tienden a laborar con su respectivo porcentaje arrojado al momento de realizar la respectiva caracterización y agrupación de los datos.

Tabla 13.

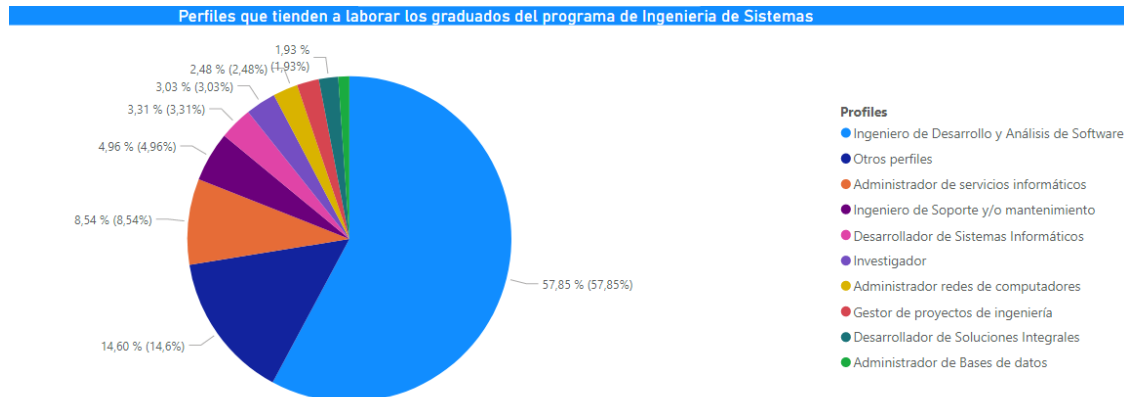
Tabla de resultados de los perfiles que tienden a laborar los graduados del programa de Ingeniería de Sistemas.

Perfiles	Porcentaje
Ingeniero de Desarrollo y Análisis de software	57.85%
Administrador de Bases de datos	1.10%
Administrador Redes de Computadores	2.48%
Ingeniero de Soporte y/o Mantenimiento	4.96%
Administrador de Servicios Informaticos	3.31%
Desarrollador de Soluciones Integrales	1.93%
Desarrollador de Sistemas Informaticos	3.31%
Investigador	3.03%
Gestor de proyectos de Ingeniería.	2.20%
Otros perfiles	14.60%

Fuente: Elaboración propia

Figura 44.

Gráfico de torta de perfiles en lo que tienden a laborar los graduados del programa de Ingeniería de Sistemas.



Fuente: Elaboración propia.

Los perfiles que más son demandados no siempre son los que cuentan con mayor porcentaje de certificación. Lo que se puede observar en la tabla 13 en comparación con la tabla 12, perfiles como el de desarrollo y análisis de software son mayores los estudiantes laborantes en esta área que los certificados. Caso contrario resulta en el perfil de administrador de base de datos donde a pesar que el porcentaje de graduados laborando en esta área, presenta un mayor índice de estudiantes certificados, mientras que, para el caso de otros perfiles, se maneja una mejor correlación.

4.4. Instituciones que tienden a elegir los graduados del programa de Ingeniería de Sistemas para continuar su evolución educativa

En la tabla 14 se visualiza las diferentes Instituciones y/o Universidades en las que los graduados del programa de Ingeniería de Sistema realizan diferentes estudios, ya sea postgrados o cursos que ayuden a su evolución educativa y de igual manera estudios extracurriculares. Por otra parte, se muestra el recuento de todas las Instituciones de educación agrupadas.

Tabla 14.

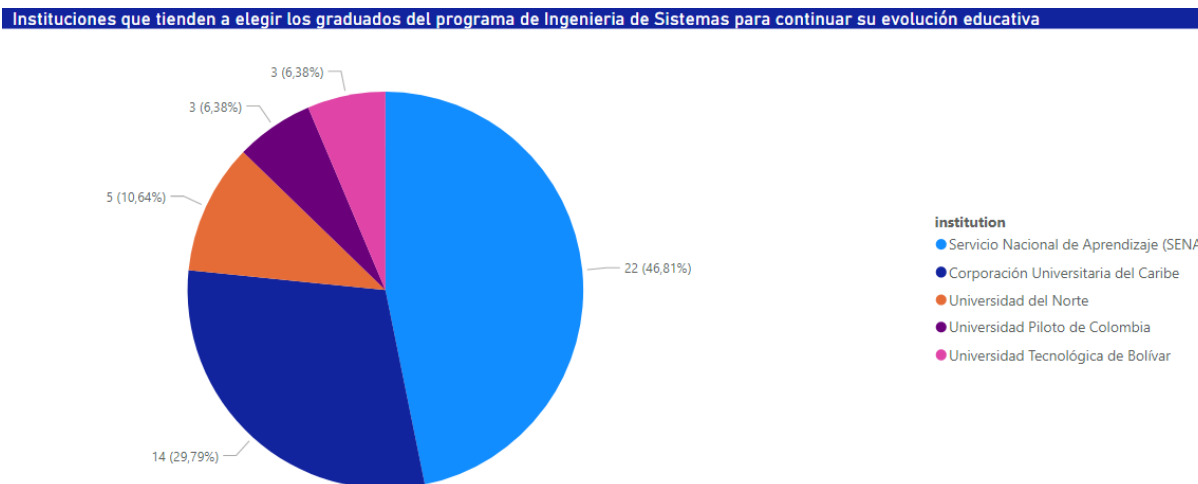
Tabla de resultados de las Instituciones que tienden a elegir los graduados del programa de Ingeniería de sistemas para continuar con su evolución educativa.

Instituciones / Universidades	Recuento
Universidad Tecnológica de Bolívar	3
Universidad Piloto Colombia	3
Universidad del Norte	5
Servicio Nacional de Aprendizaje (SENA)	22
Corporación Universitaria del Caribe	14

Fuente: Elaboración propia.

Figura 45.

Gráfico de torta que muestra el resultado de las instituciones que tienden a elegir los graduados del programa de Ingeniería de Sistemas para continuar su evolución educativa.



Fuente: Elaboración propia.

Con ayuda de la figura 45 y los datos obtenidos, se puede apreciar que los estudiantes tienden a elegir en su mayoría el Servicio Nacional de Aprendizaje (SENA) con un porcentaje del 46,81% para continuar con su proceso evolutivo. Seguido, se encuentra la institución donde realizaron sus estudios pregrados (CECAR) con un porcentaje del 29,79%.

4.5. Ubicaciones de trabajo de los graduados de Ingeniería de Sistemas

En las siguientes tablas (15 y 16) se evidencia las diferentes locaciones, ciudades y países en que los graduados del programa de Ingeniería de sistemas han desempeñado sus labores profesionales hasta el día de hoy. De igual manera, se evidencia el recuento de estas locaciones, apoyado de la sección de experiencia el cual brinda la información de las ciudades y/o países donde los graduados han realizado sus labores profesionales.

Tabla 15.

Tabla de resultados de las locaciones en las cuales estan o han estado desempeñando laboralmente su profesión los graduados del programa de Ingeniería de Sistemas.

Locación	Recuento de Ciudades donde han laborado
Sincelejo, Colombia	24
Cartagena de Indias, Colombia	2
Bogotá, Colombia	14
Medellín, Colombia	11
Montería, Colombia	2
Bauru, Sao Paulo, Brasil	3
Coveñas, Sucre, Colombia	2

Fuente: Elaboración propia

Figura 46.

Mapa donde se visualiza las ciudades y/o municipios donde actualmente desempeñan su carrera profesional o han laborado los graduados del programa de Ingeniería de Sistemas.



Fuente: Elaboración propia.

En la figura 46 se puede apreciar que, en las ciudades que más realizan sus labores los graduados del programa de Ingeniería de Sistema es en el sector norte del país, en los cuales se encuentran las ciudades: Sincelejo, Montería y Cartagena de Indias, mientras que el resto de graduados desempeñan su carrera profesional en la parte céntrica y más importantes del País de Colombia como lo son Bogotá y Medellín. Mientras que un numero pequeño desempeña o ha desempeñado sus labores en el País de Brasil, justamente en la ciudad de Sao Paulo.

Tabla 16.

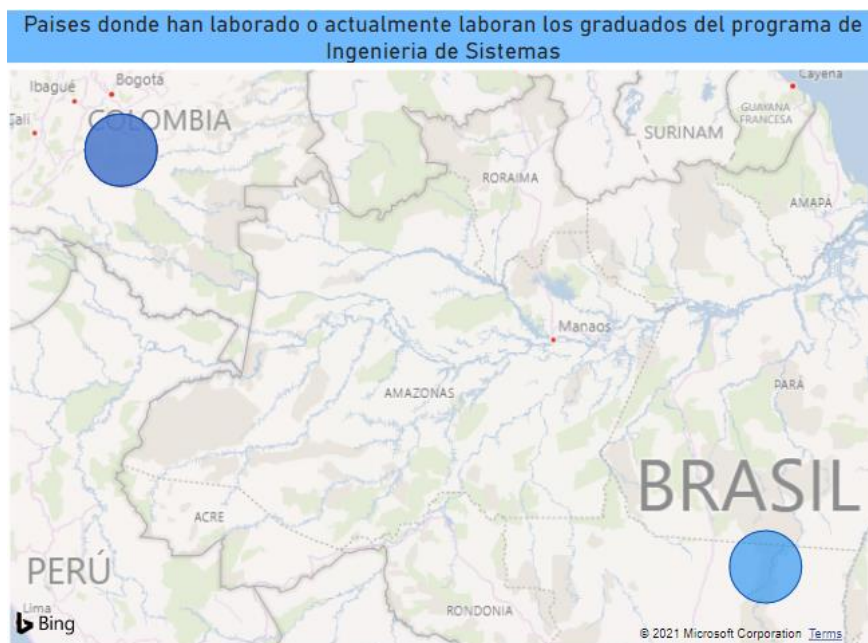
Recuento de países en los que están ejerciendo o han ejercido su perfil profesional los graduados del programa de Ingeniería de Sistemas.

País	Recuento
Colombia	55
Brasil	3

Fuente: Elaboración propia

Figura 47.

Mapa donde se visualiza los países en los cuales actualmente desempeñan su carrera profesional o han laborado los graduados del programa de Ingeniería de Sistemas.



Fuente: Elaboración propia

La figura 47 demuestra que en su mayoría los estudiantes graduados del programa de ingeniería de sistemas residen y trabajan en Colombia, sin embargo, se puede observar que un número menor han realizado labores y/o trabajan en el país de Brasil.

4.6. Perfiles que tienden a elegir los graduados para continuar con su proceso de formación en el campo de la ingeniería de sistemas

En la tabla 17, se puede observar los diferentes perfiles profesionales más específicamente en que los graduados del programa de Ingeniería de Sistemas tienden elegir para continuar sus proceso evolutivo o formativo en el campo de la Ingeniería de Sistemas, en la cual se realizó un análisis exploratorio y se evidencio patrones repetitivos que facilitó la caracterización en perfiles más específicos. De igual manera se presenta el recuento de los datos obtenidos de la sección de educación de todos los graduados.

Tabla 17.

Tabla de resultados de recuento de los perfiles que tienden a elegir los graduados para continuar con su proceso de formación en el campo de la ingeniería de Sistema.

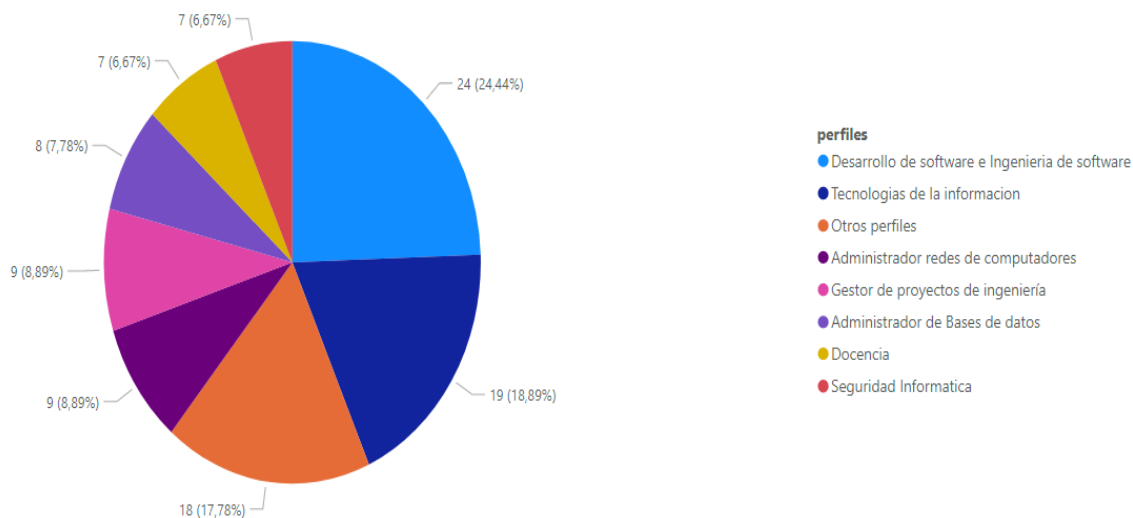
Perfiles	Recuento de perfiles
Administrador redes de computadoras	9
Administrador de Bases de datos	8
Desarrollo de software e Ingeniería de software	24
Docencia	7
Gestor de proyectos de Ingeniería	9
Seguridad informática	7
Tecnologías de la Información	19
Otros perfiles	18

Fuente: Elaboración propia

Figura 48.

Gráfico de torta en el que se visualiza los perfiles que tienden a elegir los graduados para continuar con su proceso de formación en el campo de la Ingeniería de Sistemas.

Perfiles que tienden a elegir los graduados para continuar con su proceso de formación en el campo de la ingeniería de sistemas



Fuente: Elaboración propia.

La figura 48 nos demuestra que el 24,44% del total de todos los datos obtenidos de los graduados, tienden a seguir en el campo del desarrollo de software e Ingeniería de software, seguidamente del campo de Tecnologías de la Información con un 18.89%, Mientras que el 17.78% optan otros perfiles no vinculados a la profesión cursada en la Institución.

5. Resultados Registro Calificado del Programa de Ingeniería de Sistema del año 2017

A continuación, se mostrará algunos resultados tomados desde el documento de Registro Calificado del programa de Ingeniería de Sistema del año 2017 el cual fue suministrado por el profesor y director del proyecto RAFAEL ROBERTO RUIZ ESCORCIA. Cabe aclarar que, toda la información plasmada a continuación, fue analizada y redactada por la Corporación Universitaria del Caribe apoyándose de las encuestas que esta realiza a los graduados del programa de Ingeniería de Sistema e igualmente apoyándose por la información suministrada por el Observatorio Nacional del Ministerio de Educación.

5.1. Salario

Con base a la información suministrada por el observatorio nacional se tiene que el salario promedio para el año 2015 de un ingeniero de sistemas egresado de CECAR es \$ 1.937.025, superior al promedio de enganche de recién graduados, calculado, para el mismo año, en \$ 984.000. Así mismo superior al ingreso promedio nacional para ingenieros de sistemas de \$ 1.780.102. (Ver resumen tabla 18).

Tabla 18.

Comparación de Salarios Graduados CECAR.

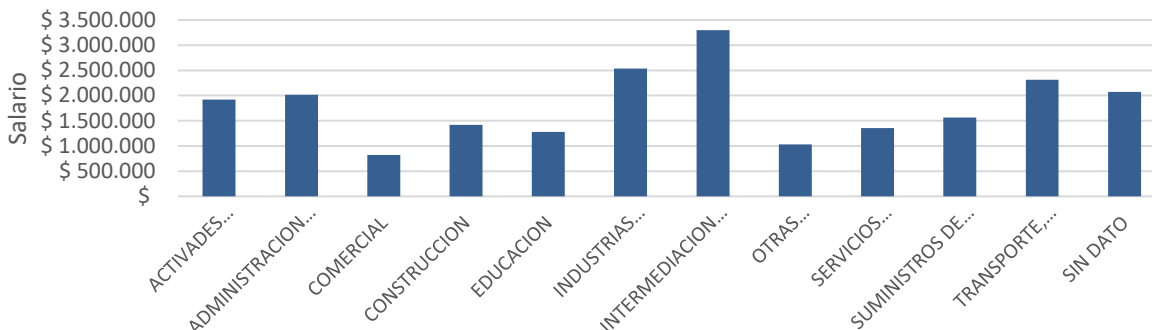
Año	Salario Enganche Recién Graduado CECAR	Salario Promedio Graduado CECAR	Salario Promedio Nacional
2015	\$ 984.000	\$ 1.937.025	1.780.102

Fuente: Tabla tomada desde el documento de Registro Calificado de Ingeniería de Sistemas del año 2017.

En cuanto a los salarios por sector económico en las diferentes regiones del país, por la fuente consultada (Observatorio laboral), se conoce que los salarios de los graduados del programa se encuentran en un rango entre un salario mínimo hasta \$3.297.800.

Figura 49.

Salario Ingreso Graduados Por Área



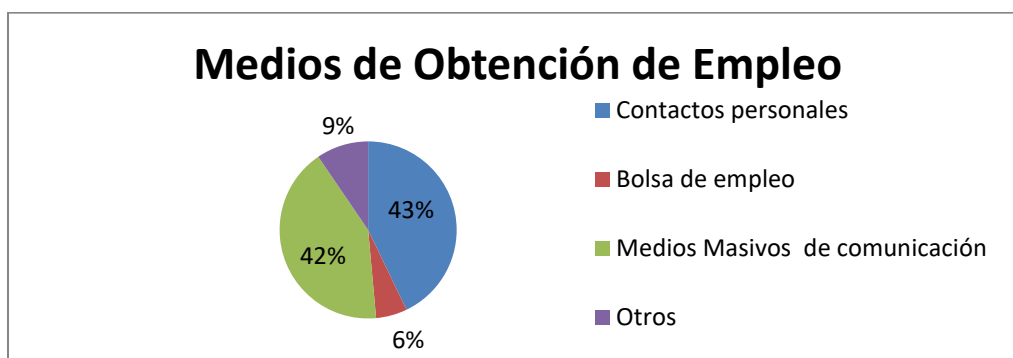
Fuente: Ilustración tomada desde el documento de Registro Calificado de Ingeniería de Sistemas del año 2017.

5.2. Medios de obtención de empleos

A través del cuestionario se consultan los principales medios por los que nuestros egresados recopilan información sobre el empleo, y se obtienen los siguientes resultados: 43% a través del contacto personal, 42% a través de medios como radio, periódicos e Internet, y en menor medida vallas publicitarias (6%) y otros medios de comunicación (9%).

Figura 50.

Medios Principales para la Obtención de Empleos



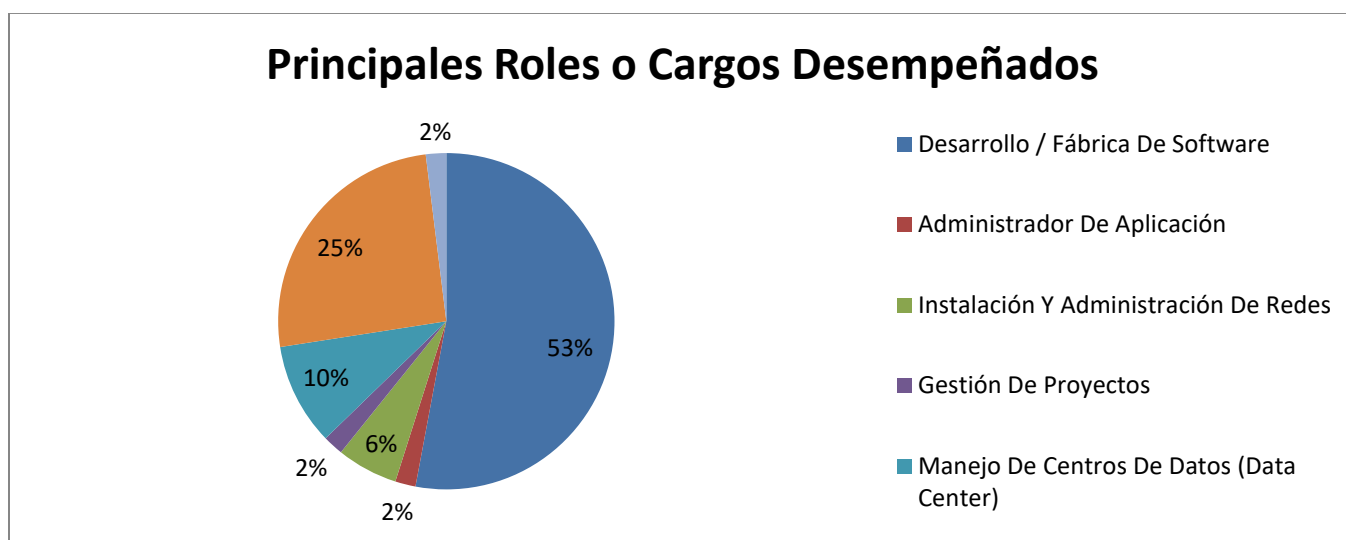
Fuente: Ilustración tomada desde el documento de Registro Calificado de Ingeniería de Sistemas del año 2017.

5.3. Principales roles o cargos desempeñados

La encuesta permitió determinar los roles o cargos que han desempeñado los Egresados después de haber obtenido el título como ingeniero de Sistemas. El 53% de los encuestados han laborado o laboran en cargos que tienen relación con el desarrollo de software. El 25% en mantenimiento o soporte de aplicativos, bien sea en sistemas de información, herramientas de software, sistemas operativos, motores de bases de datos, entre otros. Por su parte el 10% se ha dedicado a la administración de la infraestructura tecnológica de una organización (Ver figura 51).

Figura 51.

Principales roles o cargos desempeñados por los graduados del programa de Ingeniería de Sistemas.



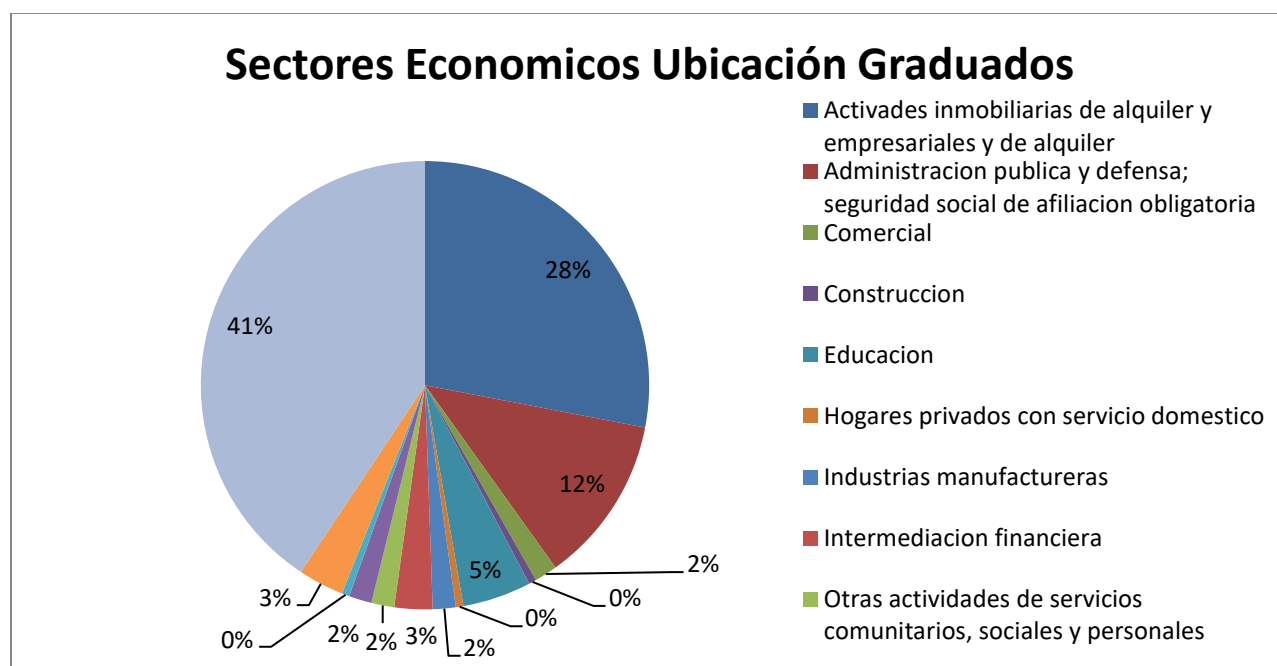
Fuente: Ilustración tomada desde el documento de Registro Calificado de Ingeniería de Sistemas del año 2017.

5.4. Sectores económicos de desempeño

En base a los datos obtenidos del observatorio nacional por la Corporación Universitaria del Caribe, muestra que el principal sector económico en el que trabajan los egresados es el de alquileres, inmuebles empresariales con un 28%. Cabe señalar que este sector se divide en las siguientes subactividades: bienes raíces, tecnología de la información y actividades relacionadas, investigación y desarrollo y otras actividades comerciales. Los graduados del programa de Ingeniería de sistemas desarrollan su actividad principal en la informática y actividades relacionadas. (Ver figura 52).

Figura 52.

Sectores Económicos Ubicación graduados



Fuente: Ilustración tomada desde el documento de Registro Calificado de Ingeniería de Sistemas del año 2017.

Teniendo en cuenta los resultados tomados desde el Registro calificado del programa de Ingeniería de Sistema del año 2017 y los resultados obtenidos desde la herramienta desarrollada, se pudo establecer que el seguimiento de los graduados tendrá una mayor cobertura para la obtención de datos con un mayor porcentaje de credibilidad para su posterior análisis.

Analizando los resultados que aportan cada técnica de recolección de datos, se puede identificar lo siguiente:

- Los datos obtenidos desde la herramienta desarrollada, aporta información relevante con respecto a qué perfiles profesionales tienden los graduados a elegir para seguir su desarrollo profesional y puesta en práctica. También a las diferentes instituciones que los graduados tienden a elegir para aumentar sus conocimientos, ya sea después de haber culminado su carrera profesional como extracurricular. De igual forma, esta herramienta ayuda a obtener información de los graduados a que sectores o perfiles de la carrera de Ingeniería de sistemas tienden a certificarse y conocer las diferentes ciudades y países que escogen los graduados para desempeñar su carrera profesional.
- Por otra parte, el Registro calificado del programa de Ingeniería de Sistema del año 2017, se puede apreciar que gracias a las encuestas y el observatorio nacional obtienen datos más profundos con respecto al salario que manejan los graduados en el sector laboral, donde este se presta para realizar diferentes análisis y comparativas ya sea nacionalmente como internacionalmente, comparar con respecto a otras universidades y entre otros. También, se puede evidenciar que este presenta datos con respecto a los sectores económicos donde laboran los graduados y estos tienden a elegir. Así mismo, se puede apreciar que estas encuestas ayudan a tener una visión más clara y más específica a los principales cargos y roles que desempeñan o han desempeñado los graduados del programa de Ingeniería de Sistemas. De igual forma, este ayuda a obtener datos de que medios de obtención de empleos los graduados tienden a conseguir oportunidades laborales.

Es por esto que, unificar los datos que extrae y clasifica la herramienta por medio de Web Scraping y con las técnicas de recolección de datos de la Institución como lo son las encuestas y el observatorio nacional, se podrían apoyar mutuamente para así complementar datos que cada una

de estas técnicas no alcanzan a abarcar, ya que, así como extraen datos en diferentes campos también cuentan con datos que tienden a ser parecidos entre estas formas de recolección de datos.

6. Conclusiones

Luego de haber conocido la problemática existente en la Corporación Universidad del Caribe CECAR, respecto a la poca recolección de datos de los graduados del programa de Ingeniería de sistemas y de igual manera el proceso con el que cuentan hasta el día de hoy de llevar el monitoreo de la evolución educativa y laboral de los graduados, se concluyó, que una posible mejora a esta solución sería el desarrollo de una herramienta que ayude a la obtención de estos datos y la caracterización de esta misma.

Al haber recopilado las necesidades que presentaba la institución con base a sus procesos de seguimiento a graduados, que debían ser ejecutados con éxito por medio de la herramienta desarrollada, se definió una metodología de desarrollo que permitiera el trabajo de un solo individuo o persona de una forma eficiente y ordenada que favorezca a que el proceso de desarrollo sea más ágil.

Gracias a las pruebas que se iban realizando a medida que se desarrollaban los diferentes Scripts, se puede concluir que también existió un código de calidad y buenas prácticas que ayude a futuro a seguir con el proceso y mejoras de esta herramienta. De igual manera, se obtuvieron los resultados que se esperaban para cada caso o acción. Como consecuencia se pudo establecer que la ocurrencia de fallas de la herramienta es mínima y que garantiza una mantenibilidad y continuo desarrollo.

Por último, luego de haber aplicado cada etapa de la metodología correctamente, encontramos una herramienta funcional para la Corporación Universitaria del Caribe que permita la extracción de datos de los graduados del programa de Ingeniería de Sistema, y que permitiera garantizar la caracterización de todos los datos para su procesamiento.

Ahora bien, desde la parte académica se fortalecieron los conocimientos adquiridos durante todo el proceso de la carrera, que gracias a los fundamentos teóricos y prácticos brindado por los docentes, se facilitó el aprendizaje de nuevos lenguajes de programación como Python y sus respectivas librerías, y de igual manera en herramientas como Selenium y Power BI los cuales fueron de gran ayuda para que este proyecto fuera realizado de la mejor manera.

Por otra parte, se realizó una investigación exhaustiva en donde se evidenció que, en toda institución de prestación de servicios educativos, como son las Universidades, uno de los procesos fundamentales es el seguimiento a los egresados o graduados de la Institución ya que esto les ayuda a identificar las fortalezas y/o limitaciones de estos para así atacar estas debilidades a los futuros graduados para así responder a las necesidades del campo laboral y la mejora continua. Se observó que la mayor parte de estas Instituciones cuentan con la realización de encuestas a estos graduados, donde esta técnica es la misma utilizada por la Corporación Universitaria del Caribe, siendo esta herramienta desarrollada, una solución innovadora para mitigar esta problemática. Esta herramienta no solo puede ayudar a la Corporación Universitaria del Caribe, sino también a cualquier Institución que cuente con esta problemática, puede lograr implementar esta solución ayudando a mitigar y a optimizar este proceso de recolección de datos.

El registro calificado del año 2017, facilitó que el proceso llevado a cabo para la extracción de los datos de los graduados permitiera la toma de decisiones al momento de obtener los datos más relevantes de cada perfil de los graduados brindado por LinkedIn. De igual forma esta herramienta desarrollada apoya la técnica con la que cuenta la Institución para la recolección de datos de los graduados en pro de mejorar los procesos de seguimiento.

Con la finalización de este proyecto, también se concluyen retos que fueron afrontados desde el proceso de extracción de datos y que estos a la vez fueron resueltos satisfactoriamente. La inexperiencia en las tecnologías y herramientas que se utilizaron ralentizó un poco el proceso de desarrollo, pero gracias a las bases que se tenían de desarrollo y las habilidades de autoaprendizaje que se nos fueron enseñadas en la Institución se pudo avanzar en el proyecto hasta llegar al punto final donde se culminó el desarrollo con estándares de calidad y gran satisfacción por haber logrado todo lo planeado.

7. Recomendaciones

Luego de haber observado los resultados y tener unas conclusiones que se relacionan claramente con la realidad que vive la institución en base a los lineamientos académicos y perfiles de interés desarrollados en el programa de Ingeniería de sistemas se recomienda que en versiones futuras o continuación de este proyecto se desarrolle un software más robusto y funcionalidades ayudaría a que este proceso fuese aún más eficiente. Dichas funcionalidades pueden ser, por ejemplo, crear un sistema con un dashboard amigable que muestre todo este proceso y poder visualizar detalladamente toda la información, como el perfil completo de un egresado, como nombre, foto, cargo actual, educación, experiencia, certificaciones y enlace hacia su perfil de LinkedIn.

También se recomienda contar con una sección de reportes el cual contenga comparativas entre diferentes variables de un grupo específico de graduados, o con sus mismas preferencias educativas o laborales para así sea más amigable y más sencillo poder analizar altos volúmenes de datos.

Por otra parte, el Web Scraping es una técnica la cual es utilizada por muchas empresas en internet, sin embargo, no deja de ser una práctica un poco polémica ya que si se utiliza de mala manera esto puede acarrear problemas al sitio web en cuestión. Alguna de esas prácticas podría ser las siguientes:

- Puede provocar perjuicios a las webs rastreadas, sobre todo si se utiliza de forma constante.
- Dependiendo de los datos extraídos, al hacer web scraping se podría estar incurriendo en competencia desleal.
- Extraer datos con intenciones maliciosas con el fin de vender esta información a otras empresas o entidades.

Pero al ser este un proyecto con intenciones educativas, no incumple o no tiene malas intenciones con respecto a los datos obtenidos desde LinkedIn, ya que estos datos no serán vendidos a terceros u obtenidos con mala intención, sino analizados para así tomar decisiones que ayuden a la institución a mejorar sus procesos de seguimientos a los graduados.

Es por esto que, si en algún momento la Institución desea seguir e implementar este proyecto, es necesario contar con la autorización por parte de los antiguos y futuros graduados para tener acceso a sus datos por medio de su perfil de LinkedIn realizando técnicas de extracción de datos para así no incumplir ninguna ley de manipulación de datos.

A partir de lo dicho anteriormente, se recomienda a la Institución gestionar espacios de capacitación, donde se le motive al estudiante a utilizar plataformas de empleabilidad que le permitan darse a conocer y crear su marca personal para que así le ayude a compartir conocimientos y relacionarse con diferentes profesionales de su ámbito académico y formativo, así mismo le facilite a la Corporación Universitaria del Caribe conocer su evolución académica y laboral para su posterior seguimiento.

Cabe destacar que, al contar con el análisis de estos datos es recomendable apoyarse de más datos que soporten este análisis con los diferentes dependencias de la Institución que lleve el control de los datos desde la fecha de ingreso a fecha de graduación de los estudiantes del programa de Ingeniería de Sistema ya que siempre se contará con datos atípicos que se tendrá que valorar con otros más específicos, por ejemplo, si un estudiante en cierto tiempo canceló la matrícula y en el transcurso de un año volvió a vincularse a la institución en su programa, esto tendrá más soporte y mejor entendimiento de la situación del estudiante en seguimiento.

Finalmente, este proyecto se inclinó específicamente al programa de Ingeniería de Sistema, pero de igual forma esta también puede ser utilizada en todos los programas educativos que la Corporación Universitaria del Caribe brinda, para así no solo llevar el seguimiento de los egresados de un solo programa, sino en general lo que ayudará a la Institución a mitigar la problemática de recolección de datos de los egresados globalmente.

Referencias Bibliográficas

- Amazon Web Services, Inc.(2020). *Bases de datos no relacionales Bases de datos de gráficos*
AWS. [online] <https://aws.amazon.com/es/nosql/>
- Bahit, E. (2012). *Scrum y eXtreme Programming para programadores*.
- Becerra, G., González, F., Reyes, J., Camargo, F. y Alfonso, Á., (2008). Seguimiento a egresados. Su importancia para las instituciones de educación superior. *Teoría y praxis investigativa*, 3 (2) <https://dialnet.unirioja.es/servlet/articulo?codigo=3701001>
- Beck, K. y Andrés, C. (s.f) *Extreme Programming Explained: Embrace Change (Kindle Location 316)*. Pearson Education. Kindle Edition. Quoted from the First Edition. <https://www.amazon.com/-/es/Kent-Beck/dp/0321278658?asin=0321278658&revisionId=&format=4&depth=1>
- Bell, J. T. (2001). *Extreme programming*.
- Borja-Buestán, C. y Cuji-Torres, V., (2013). *Metodología para la especificación de requerimientos de software*. [online] [studylib.es. https://studylib.es/doc/6622305/metodolog%C3%ADa-para-la-especificaci%C3%B3n-de-requerimientos-de-s%E2%80%A6](https://studylib.es/doc/6622305/metodolog%C3%ADa-para-la-especificaci%C3%B3n-de-requerimientos-de-s%E2%80%A6)
- Corporación Universitaria del Caribe - CECAR (2018). *Proyecto Educativo del Programa Ingeniería de Sistema*. [online] <https://cecar.edu.co/documentos/pep/pep-ingenieria-de-sistemas.pdf>
- Chacón, J., (2021). *Introducción a Pandas, la librería de Python para trabajar con datos*. [online] Profile Software Services. <https://profile.es/blog/pandas-python/>

- Cna.gov.co. 2020. Sistema Nacional De Acreditación En Colombia - CNA. <https://www.cna.gov.co/1741/article-186365.html>
- Microsoft. (2021). What is Power BI? - Power BI. [online] <https://docs.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview>
- Krasteva, I. y Ilieva, S., (2009). *Personal Extreme Programming—An Agile Process For Autonomous Developers*. ResearchGate. https://www.researchgate.net/publication/229046039_Personal_Extreme_Programming-An_Agile_Process_for_Autonomous_Developers
- Farley, E. y Pierotte, L., (2017). *Web Scraping An Emerging Data Collection Method for Criminal Justice Researchers*. 1st ed. [ebook] Justice Research and Statistics Association, p.1. <https://www.jrsa.org/pubs/factsheets/jrsa-factsheet-webscraping.pdf>
- Iyawa, G. (2020). *Personal Extreme Programming: Exploring Developers' Adoption*. In: *AMCIS 2020 Proceedings*. bepress.
- Montero, B. M., Cevallos, H. V., y Cuesta, J. D. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espiraes revista multidisciplinaria de investigación*, 2(17).
- Ojeda, J. C., y Fuentes, M. D. C. G. (2012). Taxonomía de los modelos y metodologías de desarrollo de software más utilizados. *Universidades*, (52), 37-47.
- Ortega Alba, L. and Pérez Rengifo, K., 2015. Diseño e implementación de una aplicación web para el monitoreo de egresados de ingeniería de sistemas en la universidad de córdoba utilizando georreferenciación y códigos QR. [online] <https://core.ac.uk/download/322624724.pdf>

Neoland (2019). *¿Qué es Data Science?*. <https://www.neoland.es/blog/que-es-data-science>

(2020). ¿Es legal el scraping, crawling o raspado web en España?
<https://datstrats.com/blog/scraping-es-legal-espana/>

Sirisuriya, D. S. (Noviembre, 2015). *A comparative study on web scraping*. (8)

Sommerville, I. y Alfonso Galipienso, M., (2011). *Ingeniería del software*. (7ma ed). Editorial
Madrid: Pearson Education.

Tablado, F. Ramírez, H. (2019). *¿El Web Scraping es legal?* | Blog de Protección de Datos.
[online] Grupo Atico34. <https://protecciondatos-lopd.com/empresas/web-scraping-legal/>.

Umphress, D. y Agarwal, R. (Enero, 2008). *Extreme Programming for A Single Person
Team*. *ResearchGate*

Universidades, S., (2020). *Metodologías de desarrollo de software: ¿qué son?* [online] Becas-
santander.com. [https://www.becas-santander.com/es/blog/metodologias-desarrollo-
software.html](https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html)

Anexos

Anexos 1. Reunión con la Coordinadora MARIA ANGELICA GARCIA MEDINA donde se abarcó las necesidades de la Corporación.

En este anexo se encuentra la reunión que se realizó con la Coordinadora María Angelica García Medina la cual en la fecha la cual estaba a cargo.

https://drive.google.com/file/d/1CYYFwxf_bccwZrIzANnsdgm0SmGhJ3c/view

Anexo 2. Archivo Excel de egresados de Ingeniería de Sistemas

En este anexo se encuentra la plantilla facilitada por el Ingeniero GUILLERMO HERNANDEZ HERNANDEZ el cual le fue solicitado al departamento de Sistema de la Corporación.

https://docs.google.com/spreadsheets/d/124eELkEtTgzzpyJly2g9_6I2J_zmgQBn/edit?usp=sharing&ouid=104929651640019315579&rtpof=true&sd=true

Anexo 3. Código fuente de la herramienta desarrollada

En este anexo se encuentra el enlace donde están almacenados los diferentes Scripts de la herramienta desarrollada.

<https://github.com/josbp0107/linkedin-thesis>

Anexo 4. Informe de análisis de web scraping en Power BI.

En este anexo se encuentran los diferentes reportes de los datos extraídos desde el sitio web de LinkedIn el cual fue analizado por medio de la herramienta de Microsoft Power BI y fue desplegado en la web para su consulta desde cualquier dispositivo electrónico.

<https://app.powerbi.com/view?r=eyJrIjojNjFmODU1ZTAAtZmEwOS00ZWUyLWE0ZDQtZDY4ZGU3YTAxYTVkIiwidCI6ImJhYjBiNjc5LWJkNWYtNGZlOC1iNTE2LWM2YjhiMzE3Yzc4MiIsImMiOjR9&pageName=ReportSection>