

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

TEMA:

**Prototipo de mapa de calor para identificar la demografía descriptiva de
distribución geográfica de los estudiantes de la Facultad de Ingeniería de
la Universidad Católica de Santiago de Guayaquil.**

AUTOR (ES):

Ramírez Vega, Héctor Xavier

**Trabajo de Integración curricular previo a la obtención del
título de**

INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

TUTOR:

Ing. Salazar Tovar, César Adriano, Msg

Guayaquil, Ecuador

16 de febrero del año 2023



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA

CARRERA DE CIENCIAS DE LA COMPUTACIÓN

CERTIFICACIÓN

Certificamos que el presente trabajo de integración curricular fue realizado en su totalidad por el Sr. Ramírez Vega, Héctor Xavier como requerimiento para la obtención del título de **INGENIERO EN CIENCIAS DE LA COMPUTACIÓN**.

TUTOR (A)

f. _____

Ing. Salazar Tovar, César Adriano

Guayaquil, a los 16 del mes de febrero del año 2023



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA

CARRERA DE CIENCIAS DE LA COMPUTACIÓN

DECLARACIÓN DE RESPONSABILIDAD

Yo, Ramírez Vega Héctor Xavier

DECLARO QUE:

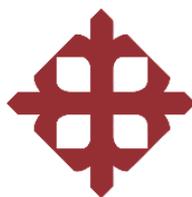
El Trabajo de Integración Curricular, “**Prototipo de mapa de calor para identificar la demografía descriptiva de distribución geográfica de los estudiantes de la facultad de Ingeniería de la Universidad Católica de Santiago de Guayaquil.**” previo a la obtención del título de **INGENIERO EN CIENCIAS DE LA COMPUTACIÓN.** ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Integración Curricular referido.

Guayaquil, a los 16 del mes de febrero del año 2023

f. _____

Ramírez Vega Héctor Xavier



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA

CARRERA DE CIENCIAS DE LA COMPUTACIÓN

AUTORIZACIÓN

Yo, **Ramírez Vega Héctor Xavier**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Integración Curricular, “**Prototipo de mapa de calor para identificar la demografía descriptiva de distribución geográfica de los estudiantes de la facultad de Ingeniería de la Universidad Católica de Santiago de Guayaquil.**”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 16 del mes de febrero del año 2023

EL AUTOR:

f. _____

Ramírez Vega Héctor Xavier



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERIA

CARRERA DE INGENIERIA EN CIENCIAS DE LA COMPUTACIÓN

REPORTE URKUND

The screenshot shows a web browser window with the URL <https://secure.arkund.com/old/view/150370845-521705>. The page header displays the URKUND logo. The main content area contains the following information:

Documento	TIC Héctor Ramírez - Documento Final.docx (D157478667)
Presentado	2023-01-31 10:49 (-05:00)
Presentado por	cesar.salazar@cu.ucsg.edu.ec
Recibido	cesar.salazar.ucsg@analysis.arkund.com
Mensaje	TIC Ramirez Mostrar el mensaje completo

0% de estas 28 páginas, se componen de texto presente en 0 fuentes.

Fecha de elaboración: 31 de enero de 2023



Ing. César Salazar Tovar, Mgs.
Tutor de Trabajo de Integración Curricular
Carrera de Computación

AGRADECIMIENTO

Quiero empezar agradeciendo al Ing. Salazar Tovar, César Adriano, Msg por no solo instruirme en mi tema del trabajo de integración curricular, también por ser un excelente docente y humilde persona que siempre estuvo dispuesto a ayudar a sus estudiantes.

Agradecer a cada uno de los docentes que prestaron su conocimiento y sabiduría para mi crecimiento personal y profesional, siempre empujándonos a ser mejores y que nos guste lo que hacemos, poniendo siempre por delante los valores éticos y morales.

Para finalizar, siempre tener presentes a mis dos amigos, hermanos que conocí al empezar este viaje y nunca me han defraudado, mis mejores amigos Erick Vélez y Diego Sánchez, ambos son amigos irremplazables y espero seguir contando con ustedes en el futuro.

DEDICATORIA

Dedico este trabajo a mis padres, Héctor Tulio Ramírez Chamba y Jhannet Del Carmen Vega Abril, este trabajo es uno de muchos regalos que pienso darles, por siempre dejarme a mis anchas, guiarme por el camino del bien, permitirme jugar videojuegos que sirvieron como base para el amor que le tengo a la tecnología, y lo más importante, darme todo el amor que un hijo se merece.

Dedicar este trabajo a mi tía Luz María Abril Moreta junto con mi abuela Mercedes, sin ellas no estaría aquí, ya que me dieron la oportunidad de estudiar en una universidad, este trabajo culminará lo que hace años, mi tía me dijo que debía terminar, quiero brindar orgullo a sus palabras, no te equivocaste tía.

También quiero mencionar a una persona muy especial, alguien a quien le quiero dedicar unas palabras, mi amigo de la infancia con quien he compartido momentos inolvidables, risas, comidas y una amistad que fue, es y siempre será irrompible. Otto Alejandro Alcívar Arias, es un honor ser amigo de alguien tan genial.



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA**

CARRERA INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

TRIBUNAL DE SUSTENTACIÓN

f. _____

ING. ANA CAMACHO CORONEL, MGS

DIRECTORA DE CARRERA

f. _____

ING. GALO CORNEJO GOMEZ, MGS

DOCENTE DE LA CARRERA

f. _____

ING. XAVIER MIRANDA RODRÍGUEZ, MGS

OPONENTE

ÍNDICE GENERAL

Contenido

RESUMEN	XIII
ABSTRACT.....	XIV
INTRODUCCIÓN.....	2
CAPÍTULO I	3
PLANTEAMIENTO DEL PROBLEMA	3
1.1 Formulación del Problema	3
1.2 Justificación e importancia	3
1.3 Delimitación	4
1.4 Objetivos	4
1.4.1 General	4
1.4.2 Específicos	4
1.5 Alcance	5
CAPÍTULO II	6
MARCO TEÓRICO.....	6
2.1 Antecedentes	6
2.2 Conceptos.....	9
2.2.1 Lenguaje de programación	9
2.2.2 Tipos de lenguajes de programación	11
2.2.2.1 Lenguaje de programación compilado.....	11
2.2.2.2 Lenguaje de programación interpretado	13
2.2.3 Principales lenguajes de programación	16
2.2.3.1 Python	16
2.2.3.2 Java.....	18
2.2.3.3 C++.....	19
2.2.3.4 C#.....	21
2.2.3.5 JavaScript.....	23
2.2.4 Preprocesamiento de Datos	25
2.2.5 Visualización de los datos	27
2.2.6 Mapa de Calor	29
2.2.7 Público Objetivo	31
2.2.8 Tecnología de Datos	32
2.2.9 Universidad Católica de Santiago de Guayaquil.....	34
CAPÍTULO III	38

METODOLOGÍA DE LA INVESTIGACIÓN	38
3.1 Tipo de investigación	38
3.2 Diseño de la investigación	41
3.3 Técnicas e instrumentos para obtención de información.....	41
3.4 Análisis de resultados	42
CAPÍTULO IV	49
PROPUESTA TECNOLÓGICA	49
4.1 Tratamiento de datos	49
4.2 Herramientas utilizadas en el Desarrollo.....	54
4.2.1 Python	54
4.2.2 Google maps.....	54
4.3 Librerías de Python utilizadas para la programación	55
4.3.1 Google maps.....	55
4.3.2 Ploty.....	56
4.3.3 PPrint.....	57
4.3.4 Pandas	58
4.3.5 Dash.....	59
4.4 Tipos de Datos	60
4.4.1 Flotantes.....	60
4.4.2 String	60
4.4.3 Dataframe.....	61
CONCLUSIONES	62
RECOMENDACIONES	63
REFERENCIAS.....	64

ÍNDICE DE FIGURAS

Figura 1 Pantalla de línea de comandos de un lenguaje de programación	11
Figura 2 Entorno de WebAssembly	13
Figura 3 Comparación entre lenguaje de programación compilado e interpretado	15
Figura 4 Pantalla de comando de Python	17
Figura 5 Pantalla de salida de Java	19
Figura 6 Pantalla de comandos de C++	21
Figura 7 C# en la herramienta de Microsoft Visual Studio Code	23
Figura 8 Pantalla de comandos en donde se utiliza JavaScript	25
Figura 9 Pasos del preprocesamiento de datos	27
Figura 10 Gráficos para visualización de datos	29
Figura 11 Ejemplo de un mapa de calor	30
Figura 12 Infografía del público objetivo	32
Figura 13 Composición de la tecnología de Datos	34
Figura 14 Mapa de la UCSG	35
Figura 15 Ubicación de la UCSG en la ciudad de Guayaquil	35
Figura 16 Ubicación de UCSG en Ranking QS	36
Figura 17 Ubicación de UCSG en Ranking de Latam	36
Figura 18 Oferta académica de Grado de la UCSG	37
Figura 19. Comparativa de Python con otros lenguajes en varios aspectos	46
Figura 20. Popularidad de Python frente a otros lenguajes	47
Figura 21. Rango de Python frente a otros lenguajes	48
Figura 22 Tratado de Datos 1	53
Figura 23 Tratado de Datos 2	53
Figura 24 Interfaz de Visual Studio con código de Python	54
Figura 25 Plataforma de Google Maps	55
Figura 26 Utilización en el código de la librería Google Maps	56
Figura 27 Utilización en el código de la librería Plotly	57
Figura 28 Visualización del mapa de calor	57
Figura 29 Utilización de la Librería PPrint	58
Figura 30 Pantalla con función DataFrame de Pandas	58
Figura 31 Dash 1	59
Figura 32 Dash 2	59
Figura 33 Dash 3	60

ÍNDICE DE TABLAS

Tabla 1 Comparación entre Python y Java.....	43
Tabla 2 Comparación entre Python y JavaScript.....	44
Tabla 3 Comparación entre Python y Perl	44
Tabla 4 Comparación entre Python y TCL.....	45
Tabla 5 Comparación entre Python y Smalltalk	45
Tabla 6 Comparación entre Python y C++.....	46

RESUMEN

El actual proyecto de integración curricular posee un objetivo primordial, el cual es proponer una herramienta de ayuda para la Universidad Católica de Santiago de Guayaquil, en la cual se pueda presentar información relacionada con el lugar de procedencia y residencia de los estudiantes de esta institución. A lo largo de la ejecución del presente trabajo se tuvo muy en cuenta conceptos relacionados con los lenguajes de programación, procesamiento y visualización de datos; así como también tener presente varios aspectos relacionados con la UCSG, que es la empresa sobre la cual se está realizando este proyecto. El método de investigación empleado fue el descriptivo, ya que para hacer un progreso fue muy necesario obtener datos de una persona cuyo trabajo incluya el hecho de hacer revisiones constantes de información de los estudiantes. Esta recopilación de datos se la realizó a través de la técnica de recolección de información conocida como entrevista. La finalidad de esto era conocer que es lo que se podía hacer con la información disponible, así como con aquella extra. El desarrollo de la propuesta tecnológica dependió en gran manera de la actualidad con respecto a los datos relacionados con el lugar de procedencia y residencia de los estudiantes de la Facultad de Ingeniería de la UCSG, así como de lo recolectado en la entrevista. Todo esto con el objetivo de poder desarrollar una herramienta tecnológica que cumpla con las necesidades presentes en la actualidad con respecto a datos geográficos de los alumnos. Con la creación de este software informático se tiene como expectativa poder ayudar a que quienes toman decisiones de peso en la UCSG, puedan hacerlo de una mejor manera teniendo de primera mano esta información de procedencia y residencia de los estudiantes.

Palabras clave: Lenguaje de programación, procesamiento de datos, visualización de datos, demografía, geografía, mapa.

ABSTRACT

The current curricular integration project has a primary objective, which is to propose a help tool for the Catholic University of Santiago de Guayaquil, in which information related to the place of origin and residence of the students of this institution can be presented. Throughout the execution of this work, concepts related to programming languages, data processing and visualization were considered; as well as to keep in mind several aspects related to the UCSG, which is the company on which this project is being carried out. The research method used was descriptive, since to make progress, it was very necessary to obtain data from a person whose job includes making constant reviews of student information. This data collection was carried out through the information collection technique known as interview. The purpose of this was to know what could be done with the available information, as well as with that extra information. The development of the technological proposal depended largely on the current situation regarding the data related to the place of origin and residence of the students of the Faculty of Engineering of the UCSG, as well as what was collected in the interview. All this with the aim of being able to develop a technological tool that meets the current needs regarding geographic data of students. With the creation of this computer software, the expectation is to be able to help those who make important decisions at UCSG, to do so in a better way, having first-hand this information on the origin and residence of the students.

Key words: Programming language, data processing, data visualization, demography, geography, map.

INTRODUCCIÓN

Desde el principio de la historia, los seres humanos sintieron una gran necesidad por saber dónde se encuentra ubicados y hacia dónde quieren ir. Como una forma muy primordial de conocer la localización se tenía la posición de los astros, como son: el sol, la luna, las estrellas y otros planetas.

Posteriormente las personas empezaron a crear mapas muy básicos con diferentes materiales como conchas de mar y palos de madera. Estos que eran hechos de esta forma empezaron a ser plasmados de una mejor manera en papiro, y luego en papel, cuando el mismo fue inventado. Después se procedió a inventar varios instrumentos de navegación como son el astrolabio y la brújula, que presentaron una forma muy eficiente de llegar a nuevos lugares y territorios. Y muy recientemente se tiene el sistema GPS, que marca la ubicación en tiempo real de los diferentes usuarios. Todos estos casos nos muestran que el ser humano siempre ha visto la forma de poder conocer dónde se encuentra y la localización de los lugares que están a su alrededor. Esto siempre ha sido de gran ayuda para poder realizar avances tanto a nivel de humanidad como a nivel de civilización.

En consecuencia, no es de extrañar que hoy en día se sigan buscando maneras de mejorar y de aprovechar al máximo las herramientas de ubicación existentes. En la actualidad los sistemas de geolocalización satelital todavía presentan algunos fallos, sin embargo, pueden ofrecer una función muy importante si son aplicados a los niveles educativos e institucionales. Las academias de educación superior tienen muy presente los orígenes de los individuos que forman parte de sí, sobre todo porque no es raro que estos vengan de lugares muy distintos. Solo por poner un ejemplo muy específico, los estudiantes que asisten a la Universidad Católica de Santiago de Guayaquil proceden de provincias como Guayas, Manabí, El Oro, Los Ríos, Santa Elena, Pichincha, entre otras. Por lo tanto, no les vendría para nada mal contar en su haber con una herramienta que permita mostrar el lugar de donde proceden sus diversos estudiantes.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 Formulación del Problema

Teniendo en cuenta la situación actual de la UCSG, siendo más específicos, el apartado de los datos que están almacenados en el sistema, los cuales y, yendo un poco más a fondo entre estos, datos como las direcciones de procedencia de los estudiantes. Estas direcciones se encuentran de una forma no legible o en su defecto, no son útiles cuando son empleadas por herramientas de mapeo, un ejemplo en estas herramientas podría ser una muy conocida como Google Maps.

Debido a ello las decisiones administrativas en diferentes procesos como admisión de estudiantes que la universidad decida realizar en cuanto al alumnado, es de vital importancia tomar en cuenta la lejanía o cercanía de los estudiantes con respecto a la universidad para posibles actividades como: planes educativos, formación de grupos de estudio, campañas publicitarias, reuniones de grupos en lugares específicos e incluso prácticas comunitarias y pre-profesionales. Pero al haber un desconocimiento de su ubicación de procedencia o de domicilio hace que sea un trabajo laborioso.

1.2 Justificación e importancia

El proyecto busca ser una base en toma de decisiones para la UCSG, si bien las direcciones están registradas y posiblemente puedan usarse manualmente, esto no es práctico ni mucho menos útil al momento de llevar a cabo procesos administrativos, no existe una herramienta visual, la cual, es mucho más entendible que muchas direcciones en registros, es un factor indispensable al momento de consultar la demografía en una institución como una universidad.

Esto acompaña a un mal uso de la información presente, como en la mayoría de negocios y empresas que manejan grandes volúmenes de

información, muchas veces es descartado el potencial que los datos brindan para mejorar a una empresa o proporcionar un nuevo punto de vista en cuanto a las acciones que realiza la empresa en un determinado espacio y tiempo.

Por lo anterior mencionado el mapa de calor que muestre la procedencia estudiantil es imprescindible y de tomar en cuenta a futuro para las diversas propuestas que la UCSG decida dictaminar al alumnado, los cuales, son su público objetivo.

1.3 Delimitación

El problema se delimita en los siguientes campos:

Campo:	Procesamiento de Datos
Área:	Visualización de datos
Aspecto:	Mejorar el conocimiento relacionado con los lugares de procedencia y de residencia de los estudiantes de la Universidad Católica de Santiago de Guayaquil
Tema:	Prototipo de mapa de calor para identificar la demografía descriptiva de distribución geográfica de los estudiantes de la facultad de ingeniería de la universidad católica de santiago de guayaquil.

1.4 Objetivos

1.4.1 General

Crear un prototipo de mapa de calor para identificar la demografía descriptiva de distribución geográfica de los estudiantes de la facultad de ingeniería de la Universidad Católica de Santiago de Guayaquil.

1.4.2 Específicos

- Obtener los datos que van a ser utilizados para la creación de la herramienta informática.

- Hacer un proceso de limpieza y preparación de los datos obtenidos anteriormente, para así poder trabajar de una forma más eficiente.
- Implementar el programa que mostrará las ubicaciones de procedencia de los estudiantes de la facultad de ingeniería de la UCSG
- Presentar los datos en un mapa de calor a través de la herramienta creada con anterioridad.

1.5 Alcance

- El proyecto se llevará a cabo en un plazo de 16 semanas.
- El área de muestreo está delimitada a la facultad de ingeniería de la Universidad Católica de Santiago de Guayaquil.
- Para el desarrollo del programa se hará uso de herramientas sin costo.
- El software no creará datos, transformará otros datos ya existentes.
- Los estudiantes serán ubicados en un mapa dentro de los límites del Gran Guayaquil, de no residir dentro no se graficará su ubicación y se encontrará en una tabla.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes

Año tras año, las organizaciones se han dado cuenta de los beneficios que proporciona el análisis de big data. Los científicos de datos y los investigadores exigen la evolución de las prácticas actuales para el procesamiento de datos sin procesar. La extracción de información automatizada es imposible del enorme depósito de datos ya que la mayoría de los datos no están estructurados. Los servicios de computación en la nube también nos han liderado con una tasa creciente de datos en la web, ya que estos servicios son rentables y fáciles de usar. Este fenómeno sin duda significa un desafío para el científico y analista de datos, por lo tanto, Big Data caracterizado como un volumen, velocidad y variedad muy altos requiere un nuevo procesamiento de alto rendimiento. El proceso de extracción de información relevante y útil de la avalancha de datos se conoce como minería de datos, que depende totalmente de la calidad de los datos. Los datos sin procesar suelen ser vulnerables a ruido, y es incompleto o inconsistente y contiene valores atípicos. Por lo tanto, estos datos deben procesarse antes de la aplicación de la minería de datos (Jamshed et al., 2019).

El preprocesamiento de datos implica la transformación del conjunto de datos sin procesar en un formato comprensible. El preprocesamiento de datos es una etapa fundamental en la minería de datos para mejorar la eficiencia de los datos. Los métodos de preprocesamiento de datos afectan directamente los resultados de cualquier algoritmo analítico; sin embargo, los métodos de preprocesamiento pueden variar según el área de aplicación. El preprocesamiento de datos es una importante etapa del proceso de minería de datos (García et al., 2016). La preparación de datos se refiere a cualquier acción destinada a aumentar la calidad, la usabilidad, la accesibilidad o la portabilidad de los datos. El objetivo final de la preparación de datos es permitir que los sistemas analíticos con datos limpios y consumibles se transformen en información procesable. El preprocesamiento de datos abarca numerosas prácticas como limpieza, integración, transformación y reducción. La fase de

preprocesamiento puede consumir una cantidad sustancial de tiempo, pero el resultado es un conjunto de datos final, que se prevé correcto y beneficioso para futuros algoritmos de minería de datos.

Los datos sin procesar disponibles en el almacén de datos, data marts, archivos de bases de datos generalmente no están organizados para el análisis, ya que pueden estar incompletos, ser inconsistentes o pueden estar distribuidos en varias tablas o representados en un formato diferente, en fin, está sucio. El proceso de descubrir conocimiento a partir de fuentes de datos cronológicos masivos se denomina descubrimiento de conocimiento en bases de datos (KDD) o Data Mining. Es la era de los grandes datos y todos los campos de la vida están generando datos a un nivel drástico. La tarea más desafiante es obtener la información correcta de las fuentes de datos actuales.

La tarea de reorganizar datos se conoce como preparación de datos. Se utiliza para descubrir el conocimiento anticipado. Incorpora la comprensión del problema basado en el dominio bajo consideración y luego una recopilación de datos específicos para lograr los objetivos previstos (Famili et al., 1997). Hasta el 80 por ciento del tiempo del analista de datos se consume en la preparación de datos. Los datos seleccionados se pueden preprocesar para la minería de datos. El preprocesamiento de datos es la mejor solución para aumentar la calidad de los datos. El preprocesamiento de datos incluye limpieza de datos, normalización de datos, transformación, extracción y selección de características, etc. Los datos procesados son el conjunto de entrenamiento para el algoritmo de aprendizaje automático.

Con respecto a la visualización de datos, esta ha existido durante siglos, y muchos estarían de acuerdo en que comenzó a finales de 1700 con William Playfair, más conocido como el "padre de los gráficos estadísticos". A Playfair se le atribuye haber inventado el gráfico de líneas, barras y pastel que usamos para a menudo hoy.

Florence Nightingale es famosa por su trabajo como enfermera durante la guerra de Crimea, pero también fue periodista de datos, conocida por sus diagramas de "coxcomb" o "rosa". Estos gráficos revolucionarios la ayudaron a

luchar por mejores condiciones en los hospitales y, en última instancia, salvaron la vida de los soldados.

Una de las visualizaciones de datos históricos más conocidas proviene de Charles Joseph Minard. Minard fue un ingeniero civil francés famoso por su representación de datos numéricos en mapas. Su obra más famosa es el mapa de la campaña rusa de Napoleón de 1812 que muestra la dramática pérdida de su ejército durante el avance sobre Moscú y la posterior retirada (Post et al., 2002).

La visualización de datos es el proceso de representar datos en un formato gráfico o pictórico. Esto permite que las personas vean relaciones y patrones que serían difíciles de discernir a partir de datos sin procesar. La visualización de datos se puede utilizar para comunicar ideas complejas de forma rápida y eficaz. Permite que las personas que nunca antes vieron números sin procesar o que incluso entendieron lo que estaban viendo en un gráfico capten rápidamente ideas complejas a través de imágenes que realmente arrojan luces sobre las relaciones entre las variables, lo que le permite comprender algo cuya importancia podría pasar desapercibida. por lectores sin algún tipo de conocimiento previo.

La visualización de datos es una herramienta poderosa que tiene muchos beneficios y puede ayudarnos a comprender mejor el mundo que nos rodea. Lo hace al permitir que los usuarios vean patrones en los datos, que de otro modo podrían pasar desapercibidos o perderse, entre otras cosas; también permite que las personas comuniquen sus hallazgos de manera clara y concisa con otras personas interesadas (ya sean investigadores que buscan respuestas ellos mismos o miembros de una audiencia) (Waskom, 2021). Y finalmente, ¡igual de importante para aquellos que trabajan en empresas cuyo trabajo implica tomar decisiones basadas en este tipo de información!, a menudo, cuando se usan gráficos como gráficos de líneas o diagramas de dispersión, la visualización de datos puede facilitar la visualización de valores atípicos que podrían ser importantes.

A medida que la "era de Big Data" se pone en marcha, la visualización es una herramienta cada vez más importante para dar sentido a los billones de filas

de datos que se generan todos los días. La visualización de datos ayuda a contar historias al seleccionar los datos en una forma más fácil de entender, destacando las tendencias y los valores atípicos. Una buena visualización cuenta una historia, eliminando el ruido de los datos y destacando información útil.

Sin embargo, no es tan fácil como simplemente adornar un gráfico para que se vea mejor o abofetear la parte de "información" de una infografía. La visualización efectiva de datos es un delicado acto de equilibrio entre forma y función. El gráfico más simple podría ser demasiado aburrido para captar cualquier aviso o decir un punto poderoso; la visualización más sorprendente podría fallar por completo al transmitir el mensaje correcto o podría decir mucho (Chen et al., 2007). Los datos y las imágenes deben trabajar juntos, y es un arte combinar un gran análisis con una gran narración.

2.2 Conceptos

2.2.1 Lenguaje de programación

Un lenguaje de programación de computadoras corresponde a cualquiera de varios lenguajes para expresar un conjunto de instrucciones detalladas para una computadora digital. Dichas instrucciones pueden ejecutarse directamente cuando están en la forma numérica específica del fabricante de la computadora conocida como lenguaje de máquina, después de un proceso de sustitución simple cuando se expresan en un lenguaje ensamblador correspondiente, o después de la traducción de algún lenguaje de "nivel superior" (Bal et al., 1989). Aunque hay muchos lenguajes de programación, relativamente pocos son ampliamente utilizados.

Aunque muchos lenguajes comparten similitudes, cada uno tiene su propia sintaxis. Una vez que un programador aprende las reglas, la sintaxis y la estructura del lenguaje, escribe el código fuente en un editor de texto o IDE. Luego, el programador a menudo compila el código en un lenguaje de máquina que la computadora puede entender. Los lenguajes de secuencias de comandos, que no requieren un compilador, utilizan un intérprete para ejecutar la secuencia de comandos (Hope, 2021).

Los lenguajes de programación se utilizan para facilitar la comunicación sobre la tarea de organizar y manipular información, y para expresar algoritmos con precisión. Algunos autores restringen el término "lenguaje de programación" a aquellos lenguajes que pueden expresar todos los algoritmos posibles; a veces, el término "lenguaje informático" se utiliza para lenguajes artificiales más limitados. Se han creado miles de lenguajes de programación diferentes y cada año se crean nuevos.

Los lenguajes de programación difieren de la mayoría de las otras formas de expresión humana en que requieren un mayor grado de precisión y exhaustividad. Cuando se usa un lenguaje natural para comunicarse con otras personas, los autores y hablantes humanos pueden ser ambiguos y cometer pequeños errores, y aun así esperar que se entienda su intención (Iverson, 1962). Sin embargo, las computadoras hacen exactamente lo que se les dice que hagan y no pueden entender el código que el programador "tenía la intención" de escribir. La combinación de la definición del lenguaje, el programa y las entradas del programa debe especificar completamente el comportamiento externo que ocurre cuando se ejecuta el programa.

Muchos idiomas se han diseñado desde cero, se han modificado para satisfacer nuevas necesidades, se han combinado con otros idiomas y, finalmente, han caído en desuso. Aunque ha habido intentos de diseñar un lenguaje informático "universal" que sirva para todos los propósitos, ninguno de ellos ha sido aceptado en esta función.

```

1  /*
2  * This line basically imports the "stdio" header file, part of
3  * the standard library. It provides input and output functionality
4  * to the program.
5  */
6  #include <stdio.h>
7
8  /*
9  * Function (method) declaration. This outputs "Hello, world\n" to
10 * standard output when invoked.
11 */
12 void sayHello(void) {
13     // printf() in C outputs the specified text (with optional
14     // formatting options) when invoked.
15     printf("Hello, world!\n");
16 }
17
18 /*
19 * This is a "main function". The compiled program will run the code
20 * defined here.
21 */
22 int main(void)
23 {
24     // Invoke the sayHello() function.
25     sayHello();
26     return 0;
27 }

```

Figura 1 Pantalla de línea de comandos de un lenguaje de programación

Tomado de “Programming Language”, del equipo de Wikipedia

2.2.2 Tipos de lenguajes de programación

2.2.2.1 Lenguaje de programación compilado

Los lenguajes compilados son lenguajes de programación que deben traducirse a instrucciones legibles por máquina mediante compiladores, que son programas que convierten el código fuente legible por humanos antes de ejecutar cualquier código. Se genera un archivo ejecutable y luego el programa compilado se pasa a la máquina de destino para su ejecución.

Los lenguajes compilados son rápidos y eficientes, ya que las instrucciones se compilan previamente en el idioma nativo de la máquina de destino, que luego no necesita asistencia mientras ejecuta las instrucciones (Ernest et al., 2017).

Algunos ejemplos de lenguajes compilados incluyen:

- C, C++ y C#
- VAMOS
- Óxido
- Haskell
- Cobol

En la lista de los diez primeros, C++ (#5) y C (#10) son lenguajes compilados de forma nativa. Rust y Go son otros dos lenguajes compilados populares que no llegaron al top 10. En general, estos lenguajes se compilan en un formato binario específico para la arquitectura y el sistema operativo. Un lenguaje compilado no necesita un intérprete o una máquina virtual para ejecutar el programa.

Los lenguajes compilados han disfrutado del soporte inicial de WebAssembly. De hecho, los cuatro (C, C++, Rust y Go) se pueden compilar en WebAssembly. Esto puede parecer sorprendente, ya que estos lenguajes generalmente se compilan por arquitectura y sistema operativo. Es decir, un programa en C por lo general debe compilarse de una manera para ejecutarse en ARM y Linux, y compilarse de una manera diferente para ejecutarse en macOS con un procesador Intel (y, a veces, el código debe cambiarse por plataforma) (Carpenter et al., 2017). Pero cuando se compilan en WebAssembly, pueden ejecutarse en cualquier tiempo de ejecución de WebAssembly, independientemente del sistema operativo, la arquitectura u otros detalles similares de bajo nivel.

Cuando se trata del soporte del compilador WebAssembly, Rust y C son los más avanzados, cada uno con un soporte impecable. El soporte de C++ viene a través de los compiladores de C y, en consecuencia, también está bien soportado. La distribución oficial de Go (propiedad de Google) está un poco atrasada, pero el proyecto TinyGo ha compensado la diferencia. Todos estos lenguajes están listos para la producción en WebAssembly.

Efectivamente, compilar uno de estos lenguajes "nativos" en WebAssembly significa que se comportan más como lenguajes de máquinas virtuales.

Entre los diez primeros, hay dos lenguajes de máquinas virtuales: Java (#2) y C# (#5). Otros lenguajes populares de máquinas virtuales incluyen Scala, F# y Kotlin. Estos lenguajes están compilados, pero no en código nativo. Se compilan en un formato intermedio llamado genéricamente "código de bytes". Luego, ese código se introduce en un tiempo de ejecución especial o una máquina virtual (JVM para Java o CLR para C#).

.NET (incluidos C#, F# y ASP.net) tiene una excelente compatibilidad con WebAssembly en el navegador y sabemos que el equipo de Microsoft está trabajando en muchas otras mejoras. El soporte de Kotlin está avanzando rápidamente y hay una serie de proyectos para Java y Scala (Schafmeister, 2016).

Dado que estos idiomas se migran a WebAssembly, ya no necesitarán sus tiempos de ejecución específicos del idioma. Por ejemplo, Kotlin no necesitará la JVM. Se compilará directamente en WebAssembly que se puede ejecutar directamente en un tiempo de ejecución de WebAssembly.

En este momento, el soporte para estos lenguajes es menos sólido que los lenguajes compilados que vimos anteriormente (aunque todos están mejorando rápidamente). Pero si el impulso continúa como hasta ahora, pronto veremos muchos lenguajes de máquinas virtuales compilados en WebAssembly.

```
(module
  (type $t0 (func (param i32 i32 i32) (result i32)))
  (type $t1 (func (param i32)))
  (type $t2 (func (param i32 i32 i32 i32) (result i32)))
  (type $t3 (func (param i32 i32) (result i32)))
  (type $t4 (func (param i32 i32 i32 i32 i32 i32) (result i32)))
  (type $t5 (func))
  (type $t6 (func (result i32)))
  (type $t7 (func (param i32) (result i32)))
  (type $t8 (func (param i32 i64 i32) (result i64)))
  (import "env" "putc_js" (func $putc_js (type $t1)))
  (import "env" "__syscall13" (func $__syscall13 (type $t2)))
  (import "env" "__syscall11" (func $__syscall11 (type $t3)))
  (import "env" "__syscall15" (func $__syscall15 (type $t4)))
  (func $__wasm_call_ctors (type $t5))
  (func $main (export "main") (type $t6) (result i32)
    i32.const 1024
    call $putc
    drop
    i32.const 0)
  (func $write_c (export "write_c") (type $t0) (param $p0 i32) (param $p1 i32) (param $p2 i32) (result i32)
```

Figura 2 Entorno de WebAssembly

Tomado de “A Primer on WebAssembly” de Divya Mohan, 2022

2.2.2.2 Lenguaje de programación interpretado

Los lenguajes interpretados son lenguajes de programación para los cuales las instrucciones no están precompiladas para la máquina de destino en una forma legible por máquina. Más bien, estos idiomas son asistidos por un intérprete. Un intérprete es un programa que traduce código fuente legible por humanos de alto nivel en código de destino legible por máquina de bajo nivel línea por línea mientras se ejecuta el programa interpretado (Bhaskar et al., 2022).

Interpretar el idioma es menos eficiente porque el intérprete debe estar presente durante todo el proceso, pero estos idiomas también son altamente adaptables.

Algunos ejemplos de lenguajes interpretados incluyen:

- Python
- JavaScript
- PHP
- MATLAB
- Perl
- Ruby

Cinco lenguajes de secuencias de comandos llegaron a la lista de los diez principales lenguajes de programación: JavaScript (n.º 1), Python (n.º 2), PHP (n.º 4), TypeScript (n.º 8) y Ruby (n.º 9). Los lenguajes de secuencias de comandos normalmente no se compilan. En su lugar, el código se introduce directamente en un intérprete en tiempo de ejecución. El intérprete lee y ejecuta el programa sobre la marcha. Debido a que el código se carga y ejecuta en tándem, los entornos de lenguaje de secuencias de comandos a veces hacen cosas que los lenguajes compilados no pueden. Por ejemplo, es trivial incluir una nueva pieza de código fuente en tiempo de ejecución, mientras que esa no es una característica de la mayoría de los lenguajes compilados.

Mientras que el trabajo en el soporte de WebAssembly para lenguajes compilados se ha centrado en compilar los propios programas en WebAssembly, el trabajo con lenguajes de secuencias de comandos se ha centrado en compilar el tiempo de ejecución de secuencias de comandos en WebAssembly (Courtney, 1995). En este modelo, el motor se carga en el tiempo de ejecución de WebAssembly y luego los scripts se cargan en el motor. El modelo de ejecución de dos niveles que presenta es más complicado que los lenguajes compilados. Pero al apegarse a lo familiar, este modelo también es el más fácil para los desarrolladores en ese lenguaje. Por ejemplo, no necesitan aprender a compilar su código previamente interpretado.

De los lenguajes de secuencias de comandos entre los diez primeros, JavaScript es el que se encuentra en desarrollo más activo. En particular, el motor SpiderMonkey que impulsa la implementación de JavaScript de Firefox ahora se compila en WebAssembly.

A partir de enero de 2022, tanto Ruby como Python ahora pueden compilar sus respectivos tiempos de ejecución en WebAssembly. Si bien el soporte para Wasm es temprano en ambos, creemos que madurarán muy rápidamente ahora que la parte difícil está hecha.

TypeScript generalmente se traduce a JavaScript. Luego, el archivo generado se introduce en el intérprete de JavaScript. Entonces, el éxito de JavaScript se acumula automáticamente en TypeScript. Pero los desarrolladores de TypeScript tienen una alternativa interesante. AssemblyScript es un superconjunto de TypeScript que se compila en WebAssembly. Si bien mantiene la sensación de un lenguaje de secuencias de comandos, en última instancia se comporta como un lenguaje compilado (Sanner, s/f).

Finalmente, PHP está ligeramente por detrás de los otros lenguajes de secuencias de comandos en cuanto a soporte. Algunos proyectos no oficiales han agregado soporte limitado para ejecutar PHP en el navegador al cargar un intérprete de PHP compilado por WebAssembly en el navegador y ejecutar el código allí. Algunos de los ejemplos son emocionantes. PHP, con su modelo de solicitud/respuesta, sería un lenguaje excelente para crear aplicaciones en la nube de WebAssembly.

Compiled		Interpreted	
PROS	CONS	PROS	CONS
ready to run	not cross platform	cross-platform	interpreter required
often faster	inflexible	simpler to test	often slower
source code is private	extra step	easier to debug	source code is public

Figura 3 Comparación entre lenguaje de programación compilado e interpretado

Tomado de “Source Code And Language Differences” de Laurende Bradford,
2014

2.2.3 Principales lenguajes de programación

2.2.3.1 Python

Python es un lenguaje de programación de alto nivel interpretado, orientado a objetos y con semántica dinámica. Sus estructuras de datos integradas de alto nivel, combinadas con la escritura y el enlace dinámicos, lo hacen muy atractivo para el desarrollo rápido de aplicaciones, así como para su uso como lenguaje de secuencias de comandos o pegamento para conectar componentes existentes entre sí. La sintaxis simple y fácil de aprender de Python enfatiza la legibilidad y, por lo tanto, reduce el costo de mantenimiento del programa. Python admite módulos y paquetes, lo que fomenta la modularidad del programa y la reutilización del código (Hao & Ho, 2019). El intérprete de Python y la extensa biblioteca estándar están disponibles en formato fuente o binario sin cargo para todas las plataformas principales y se pueden distribuir gratuitamente.

A menudo, los programadores se enamoran de Python debido a la mayor productividad que proporciona. Dado que no hay un paso de compilación, el ciclo de edición, prueba y depuración es increíblemente rápido. La depuración de programas de Python es fácil: un error o una entrada incorrecta nunca provocarán una falla de segmentación. En cambio, cuando el intérprete descubre un error, genera una excepción. Cuando el programa no detecta la excepción, el intérprete imprime un seguimiento de la pila. Un depurador a nivel de código fuente permite la inspección de variables locales y globales, la evaluación de expresiones arbitrarias, el establecimiento de puntos de interrupción, el paso por el código línea por línea, etc (Helmus & Collis, 2016). El depurador está escrito en Python mismo, lo que demuestra el poder introspectivo de Python. Por otro lado, a menudo la forma más rápida de depurar un programa es agregar algunas declaraciones de impresión a la fuente: el ciclo rápido de edición, prueba y depuración hace que este enfoque simple sea muy efectivo.

Python es popular por varias razones. Aquí hay una mirada más profunda a lo que lo hace tan versátil y fácil de usar para los codificadores(Coursera Team, 2022).

- Tiene una sintaxis simple que imita el lenguaje natural, por lo que es más fácil de leer y comprender. Esto hace que sea más rápido crear proyectos y mejorarlos.
- Es versátil. Python se puede usar para muchas tareas diferentes, desde desarrollo web hasta aprendizaje automático.
- Es amigable para principiantes, lo que lo hace popular entre los codificadores de nivel de entrada.
- Es de código abierto, lo que significa que es de uso y distribución gratuitos, incluso con fines comerciales.
- El archivo de módulos y bibliotecas de Python (paquetes de código que los usuarios de terceros han creado para expandir las capacidades de Python) es enorme y está creciendo.
- Python tiene una comunidad grande y activa que contribuye al grupo de módulos y bibliotecas de Python, y actúa como un recurso útil para otros programadores. La amplia comunidad de soporte significa que si los codificadores se encuentran con un obstáculo, encontrar una solución es relativamente fácil; seguro que alguien se ha encontrado con el mismo problema antes.

```
1 #modeling the person class
2 class Person():
3 #method to initialize name and age attributes.
4     def __init__(self,name, age):
5         self.name = name
6         self.age = age
7 #method to demonstrate what a person eats
8     def eat(self):
9         print(self.name.title() + "eats Matooke and rice")
10        print("She is"+ str(self.age) + " years old")
11    def drink(self):
12        print("Drinks water")
13 #instantiating a class.
14 my_sister = Person("Haniifa", 30)
15 #Accessing the class method through the class object.
16 my_sister.eat()
17
```

Figura 4 Pantalla de comando de Python

Tomado de “Python Programming Language”, de Nakayiza Shamim, 2019

2.2.3.2 Java

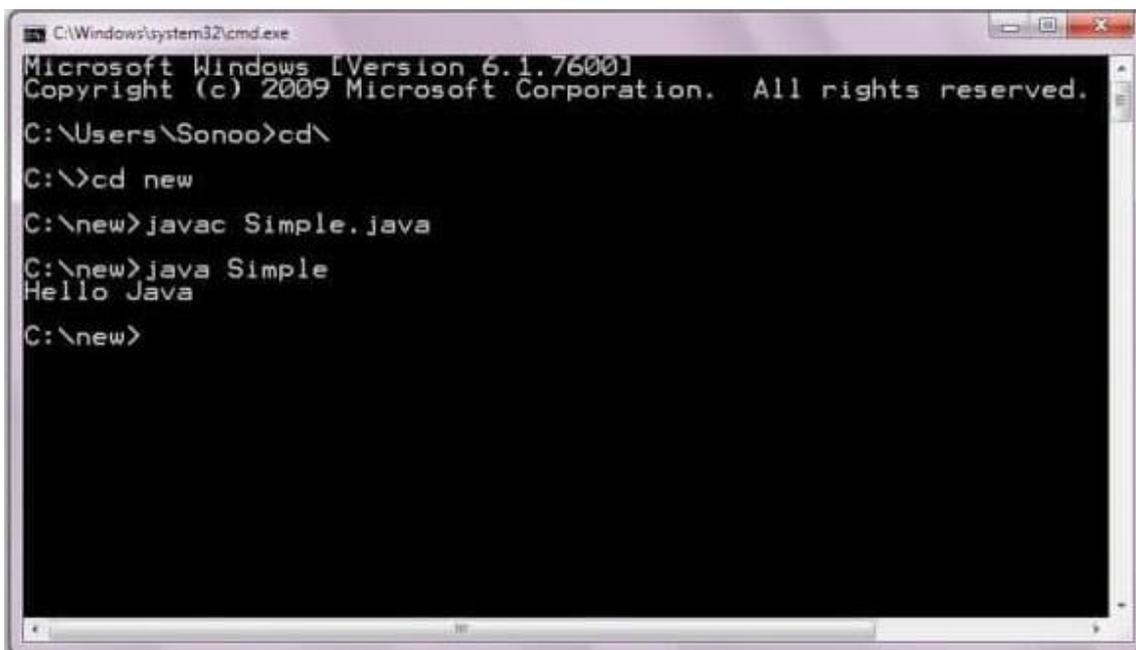
Java es un lenguaje de programación de propósito general, basado en clases y orientado a objetos, diseñado para tener menores dependencias de implementación. Es una plataforma informática para el desarrollo de aplicaciones. Java es rápido, seguro y confiable, por lo tanto. Es ampliamente utilizado para desarrollar aplicaciones Java en computadoras portátiles, centros de datos, consolas de juegos, supercomputadoras científicas, teléfonos celulares, etc (Hartman, 2020).

Java es extremadamente portátil. La misma aplicación Java se ejecutará de manera idéntica en cualquier computadora, independientemente de las características del hardware o del sistema operativo, siempre que tenga un intérprete de Java. Además de la portabilidad, otra de las ventajas clave de Java es su conjunto de características de seguridad que protegen una PC que ejecuta un programa Java no solo de problemas causados por códigos erróneos sino también de programas maliciosos (como virus) (Bracha, 2004). Puede ejecutar con seguridad un subprograma Java descargado de Internet, porque las características de seguridad de Java evitan que este tipo de subprogramas accedan al disco duro de una PC o a las conexiones de red. Un subprograma suele ser un pequeño programa Java que está incrustado en una página HTML.

Java se puede considerar tanto un lenguaje compilado como interpretado porque su código fuente se compila primero en un código de bytes binario. Este código de bytes se ejecuta en la máquina virtual de Java , que suele ser un intérprete basado en software. El uso de código de bytes compilado permite que el intérprete sea pequeño y eficiente. Además, este código de bytes le da a Java su portabilidad: se ejecutará en cualquier JVM que se implemente correctamente, independientemente de la configuración del hardware o software de la computadora. La mayoría de los navegadores web contienen una JVM para ejecutar applets de Java.

En comparación con C++, el código Java se ejecuta un poco más lento, pero es más portátil y tiene características de seguridad mucho mejores. La

máquina virtual proporciona aislamiento entre un programa Java que no es de confianza y la PC que ejecuta el software. La sintaxis de Java es similar a la de C++ pero los lenguajes son bastante diferentes. Por ejemplo, Java no permite a los programadores implementar la sobrecarga de operadores mientras que C++ sí lo hace. Además, Java es un lenguaje dinámico en el que puede modificar un programa de forma segura mientras se ejecuta, mientras que C++ no lo permite (Arnold et al., 2005). Esto es especialmente importante para las aplicaciones de red que no pueden permitirse ningún tiempo de inactividad. Además, todos los tipos de datos básicos de Java están predefinidos y no dependen de la plataforma, mientras que algunos tipos de datos pueden cambiar con la plataforma utilizada en C o C++.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Sonoo>cd\
C:\>cd new
C:\new>javac Simple.java
C:\new>java Simple
Hello Java
C:\new>
```

Figura 5 Pantalla de salida de Java

Tomado de “First Java Program | Hello World Example” del equipo de Javatpoint

2.2.3.3 C++

C++ es un lenguaje informático orientado a objetos creado por el destacado científico informático Bjarne Stroustrup como parte de la evolución de la familia de lenguajes C. Fue desarrollado como una mejora multiplataforma de

C para proporcionar a los desarrolladores un mayor grado de control sobre la memoria y los recursos del sistema.

Algunos llaman a C++ "C con clases" porque introduce principios de programación orientada a objetos, incluido el uso de clases definidas, en el marco del lenguaje de programación C . Con el tiempo, C ++ se ha mantenido como un lenguaje muy útil no solo en la programación de computadoras en sí, sino también para enseñar a los nuevos programadores cómo funciona la programación orientada a objetos (Stroustrup, 1999). Sin embargo, no solo admite orientación a objetos, sino también procedimental y funcional. Gracias a su alta flexibilidad y escalabilidad, C++ se puede utilizar para desarrollar una amplia gama de software, aplicaciones, navegadores, interfaces gráficas de usuario, sistemas operativos y juegos.

Hoy en día, C ++ sigue siendo muy apreciado por su notable portabilidad que permite a los desarrolladores crear programas que pueden ejecutarse en diferentes sistemas operativos o plataformas con mucha facilidad. A pesar de ser un lenguaje de alto nivel, dado que C ++ todavía está cerca de C, puede usarse para la manipulación de bajo nivel debido a su estrecha relación con el lenguaje de máquina.

Cuando C++ era nuevo, la programación orientada a objetos acababa de entrar en escena. Este tipo revolucionario de programación informática transformó el mundo de la codificación con su promesa de objetos y tipos de datos virtuales más sofisticados.

En la programación orientada a objetos, un objeto es un tipo de datos que tiene datos y funciones inherentes a su diseño (Rassokhin, 2020). Antes del advenimiento de la programación orientada a objetos, los programadores generalmente veían una base de código compuesta por instrucciones de línea de comando individuales. La identificación de objetos con datos y funciones integradas condujo a una nueva forma de empaquetar y automatizar el trabajo del código.

Para un excelente ejemplo de programación orientada a objetos en C++, una de las características más notables y útiles del lenguaje fue la pila de C++.

La pila de C++ es una clase en C++ que tiene las siguientes características: es un contenedor de almacenamiento secuencial virtual de último en entrar, primero en salir que tiene un conjunto definido de elementos. Las funciones 'push' y 'pop' empujan un nuevo elemento al final de la pila o sacan el primer elemento disponible de la parte superior de la pila (Buttice, 2021).

Los programadores han utilizado la pila de C++ de muchas maneras diferentes para lograr objetivos relacionados con la evaluación de variables y operaciones funcionales dentro de una base de código.

El lenguaje también aplica principios de encapsulación, que identifica modelos de uso y herencia, donde una clase puede heredar ciertos atributos o propiedades de otra.

```
// Carbon:
package Geometry api;
import Math;

class Circle {
    var r: f32;
};

fn PrintTotalArea(circles: Slice(Circle)) {
    var area: f32 = 0;
    for (c: Circle in circles) {
        area += Math.Pi * c.r * c.r;
    }
    Print("Total area: {0}", area);
}

fn Main() -> i32 {
    // A dynamically sized array, like `std::vector`.
    var circles: Array(Circle) = ({.r = 1.0},
                                  {.r = 2.0});
    // Implicitly converts `Array` to `Slice`.
    PrintTotalArea(circles);
    return 0;
}
```

Figura 6 Pantalla de comandos de C++

Tomado de “A new programming language that might be successor to C++”, de Pritiam Bordoloi, 2022

2.2.3.4 C#

C# es un lenguaje de programación orientado a objetos de Microsoft que tiene como objetivo combinar la potencia informática de C++ con la facilidad de

programación de Visual Basic. C# se basa en C++ y contiene funciones similares a las de Java.

C# está diseñado para funcionar con la plataforma .NET de Microsoft. El objetivo de Microsoft es facilitar el intercambio de información y servicios a través de la Web y permitir a los desarrolladores crear aplicaciones altamente portátiles (Hejlsberg et al., 2008). C# simplifica la programación mediante el uso del lenguaje de marcado extensible y el protocolo simple de acceso a objetos (SOAP), que permiten el acceso a un objeto o método de programación sin que el programador tenga que escribir código adicional para cada paso. Debido a que los programadores pueden construir sobre el código existente, en lugar de duplicarlo repetidamente, se espera que C# haga que sea más rápido y menos costoso lanzar nuevos productos y servicios al mercado.

Microsoft está colaborando con ECMA, el organismo de estándares internacionales, para crear un estándar para C#. El reconocimiento de la Organización Internacional de Normalización (ISO) para C# animaría a otras empresas a desarrollar sus propias versiones del lenguaje. Las empresas que ya utilizan C# incluyen Apex Software, Bunka Orient, Component Source, devSoft, FarPoint Technologies, LEAD Technologies, ProtoView y Seagate Software.

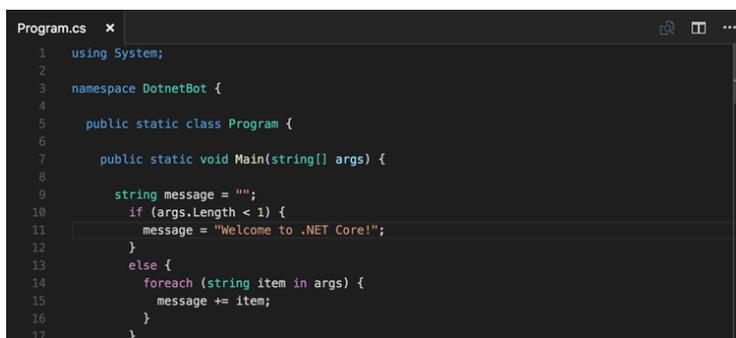
El propósito de C# fue desarrollar un lenguaje de programación que no solo sea fácil de aprender, sino que también admita la funcionalidad moderna para todo tipo de desarrollo de software (Chand, 2020).

Si se observa la historia de los lenguajes de programación y sus características, cada lenguaje de programación fue diseñado para un propósito específico para resolver una necesidad específica en ese momento.

Sin embargo, el lenguaje C# fue diseñado para tener en cuenta las necesidades comerciales y empresariales. El lenguaje C# fue diseñado para que las empresas creen todo tipo de software mediante el uso de un solo lenguaje de programación.

C# proporciona y colecciones, expresiones lambda, programación dinámica, programación asíncrona, tuplas, coincidencia de patrones, depuración avanzada y manejo de excepciones, y más (Hejlsberg et al., 2003).

Las sintaxis del lenguaje C# están influenciadas por C++, Java, Pascal y algunos otros lenguajes que son fáciles de adoptar. C# también evita la complejidad y las funciones de lenguaje no estructurado.

A screenshot of the Visual Studio Code editor showing a C# file named Program.cs. The code is as follows:

```
1 using System;
2
3 namespace DotnetBot {
4
5     public static class Program {
6
7         public static void Main(string[] args) {
8
9             string message = "";
10            if (args.Length < 1) {
11                message = "Welcome to .NET Core!";
12            }
13            else {
14                foreach (string item in args) {
15                    message += item;
16                }
17            }
18        }
19    }
20 }
```

Figura 7 C# en la herramienta de Microsoft Visual Studio Code

Tomado de “Working with C#”, del Equipo de Visual Studio Code de Microsoft

2.2.3.5 JavaScript

JavaScript (a menudo abreviado como JS) es un lenguaje ligero, interpretado y orientado a objetos con funciones de primera clase, y es mejor conocido como el lenguaje de secuencias de comandos para páginas web, pero también se usa en muchos entornos que no son de navegador. Es un lenguaje de secuencias de comandos multiparadigma basado en prototipos que es dinámico y admite estilos de programación funcionales, imperativos y orientados a objetos.

JavaScript se ejecuta en el lado del cliente de la web, que se puede usar para diseñar/programar cómo se comportan las páginas web ante la ocurrencia de un evento (Wirfs-Brock & Eich, 2020). JavaScript es un lenguaje de secuencias de comandos fácil de aprender y también potente, ampliamente utilizado para controlar el comportamiento de las páginas web.

Contrariamente a la idea errónea popular, JavaScript no es "Java interpretado". En pocas palabras, JavaScript es un lenguaje de secuencias de comandos dinámico que admite la construcción de objetos basados en

prototipos. La sintaxis básica es intencionalmente similar tanto a Java como a C++ para reducir la cantidad de conceptos nuevos necesarios para aprender el lenguaje. Las construcciones del lenguaje, como declaraciones if, bucles for y while, y los bloques switch e try... catch funcionan igual que en estos lenguajes.

JavaScript puede funcionar como un lenguaje procedimental y orientado a objetos. Los objetos se crean mediante programación en JavaScript, adjuntando métodos y propiedades a objetos vacíos en tiempo de ejecución, a diferencia de las definiciones de clases sintácticas comunes en lenguajes compilados como C++ y Java (Theisen, 2019). Una vez que se ha construido un objeto, se puede utilizar como modelo (o prototipo) para crear objetos similares.

Las capacidades dinámicas de JavaScript incluyen la construcción de objetos en tiempo de ejecución, listas de parámetros variables, variables de funciones, creación de secuencias de comandos dinámicas, introspección de objetos y recuperación de código fuente.

JavaScript es el lenguaje de programación más utilizado por las siguientes razones (Boyd, 2022):

- Universalidad
- Fácil de aprender
- Versatilidad
- Comunidad y recursos

```
<html>
<head>
  <div>
    <div>
      <form method="post" action="#" id="formvalue" onkeyup="
drawChart()" />
    </form>
  </div>
</div>

<script type="text/javascript" src="https://www.google.com/jsapi"></
script>
<script type="text/javascript">

var bid = 43;
var ask = 21;

google.load("visualization", "1", {packages:["corechart"]});
google.setOnLoadCallback(drawChart);
function drawChart() {
  var data = google.visualization.arrayToDataTable([
    ['Price', 'Quantity'],
    ['Value #1', bid],
    ['Value #2', ask],
  ]);
}
```

Figura 8 Pantalla de comandos en donde se utiliza JavaScript

Tomado de "JavaScript", del equipo de Wikipedia

2.2.4 Preprocesamiento de Datos

El preprocesamiento de datos es un paso en el proceso de extracción y análisis de datos que toma datos sin procesar y los transforma en un formato que puede ser entendido y analizado por computadoras y aprendizaje automático.

Los datos sin procesar del mundo real en forma de texto, imágenes, video, etc., son desordenados. No solo puede contener errores e inconsistencias, sino que a menudo está incompleto y no tiene un diseño regular y uniforme (Kamiran & Calders, 2012).

A las máquinas les gusta procesar información agradable y ordenada: leen los datos como 1 y 0. Por lo tanto, calcular datos estructurados, como números enteros y porcentajes, es fácil. Sin embargo, los datos no estructurados, en forma de texto e imágenes, primero deben limpiarse y formatearse antes del análisis.

Al usar conjuntos de datos para entrenar modelos de aprendizaje automático, a menudo escuchará la frase "basura que entra, basura que sale"

(García et al., 2015). Esto significa que si usa datos malos o "sucios" para entrenar su modelo, terminará con una mala, modelo mal entrenado que en realidad no será relevante para su análisis.

Los datos buenos y preprocesados son incluso más importantes que los algoritmos más poderosos, hasta el punto de que los modelos de aprendizaje automático entrenados con datos incorrectos podrían ser perjudiciales para el análisis que está tratando de hacer, brindándole resultados "basura".

Dependiendo de sus técnicas y fuentes de recopilación de datos, puede terminar con datos que están fuera de rango o incluyen una característica incorrecta, como un ingreso familiar por debajo de cero o una imagen de un conjunto de "animales del zoológico" que en realidad es un árbol. Su conjunto podría tener valores o campos faltantes. O los datos de texto, por ejemplo, a menudo tendrán palabras mal escritas y símbolos irrelevantes, URL, etc.

Cuando preprocesa y limpia correctamente sus datos, se preparará para procesos posteriores mucho más precisos. A menudo escuchamos sobre la importancia de la "toma de decisiones basada en datos", pero si estas decisiones están impulsadas por datos incorrectos, son simplemente malas decisiones.

Tareas principales en el preprocesamiento de datos (Anunaya, 2021):

- Limpieza de datos
- Integración de datos
- Reducción de datos
- Transformación de datos

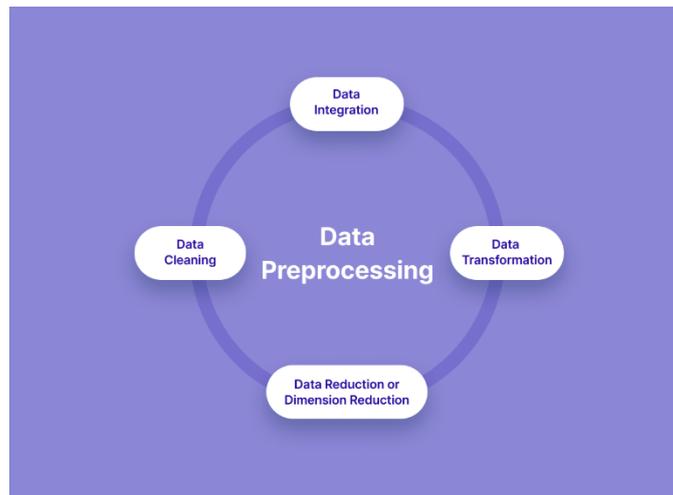


Figura 9 Pasos del preprocesamiento de datos

Tomado de “A Simple Guide to Data Preprocessing in Machine Learning”, de Pragati Baheti, 2022

2.2.5 Visualización de los datos

La visualización de datos es la representación gráfica o visual de los datos. Ayuda a resaltar los conocimientos más útiles de un conjunto de datos, lo que facilita la detección de tendencias, patrones, valores atípicos y correlaciones.

Se puede imaginar que se presenta una hoja de cálculo que contiene filas y filas de datos. Probablemente no podrá descifrar los datos sin profundizar en ellos, y es poco probable que se pueda detectar tendencias y patrones a primera vista. Ahora se puede imaginar ver los mismos datos presentados como un gráfico de barras o en un mapa codificado por colores. Es mucho más fácil ver lo que dicen los datos (Friendly, 2008).

Un gráfico de barras que muestra las calorías quemadas cada día en el transcurso de una semana, y una variante de un gráfico circular que muestra qué proporción del tiempo total de ejercicio tuvo lugar en la zona de "quema de grasa" y en las zonas de "cardio y pico"

Ese es el objetivo de la visualización de datos. Hace que los conocimientos sean visibles a simple vista, de modo que prácticamente cualquier persona pueda ver y comprender lo que está sucediendo. Cuando se hace bien,

la visualización de datos cuenta una historia. Este aspecto narrativo es crucial ya que hace que sus datos sean procesables. Hay una gran diferencia entre simplemente tener una gran cantidad de datos y comprender realmente cómo usarlos para impulsar acciones y decisiones, y la visualización de datos cierra esa brecha(Stevens, 2021).

Hay dos grandes categorías de visualización de datos: exploración y explicación.

La importancia de una visualización de datos eficaz radica en la importancia del análisis de datos en general. Vivimos en un mundo cada vez más rico en datos; a principios de 2020, el universo digital comprendía aproximadamente 44 zettabytes de datos. En perspectiva, un zettabyte es aproximadamente igual a un billón de gigabytes. Para 2025, se estima que se crearán alrededor de 463 exabytes de datos cada 24 horas en todo el mundo. Un exabyte equivale a mil millones de gigabytes. Básicamente, estamos produciendo toneladas y toneladas de datos todo el tiempo.

El análisis de datos nos permite dar sentido a (al menos parte de) esos datos. Desde una perspectiva empresarial, permite a las empresas aprender del pasado y planificar el futuro. En campos como el de la salud, puede ayudar a mejorar la atención y el tratamiento de los pacientes (Aparicio & Costa, 2015). En finanzas y seguros, puede ayudar a evaluar el riesgo y combatir la actividad fraudulenta. Esencialmente, necesitamos análisis de datos para tomar decisiones inteligentes, y la visualización de datos es una parte crucial de eso.

La visualización de datos nos ayuda a comprender lo que nos dicen ciertos datos, presentándolos de una manera accesible para una variedad de audiencias, no solo para expertos en datos. Es cómo cierra la brecha entre su experiencia como analista de datos o científico de datos y aquellas personas que pueden usar o actuar sobre los conocimientos que descubre.



Figura 10 Gráficos para visualización de datos

Tomado de “The Top 10 Types of Data Visualization Made Simple”, del equipo de Boost Labs, 2019

2.2.6 Mapa de Calor

Un mapa de calor es una representación visual de datos, que muestra valores numéricos como esquemas de color. Permite a las marcas monitorear y registrar todas las acciones que los visitantes realizan en un sitio web en función de los movimientos del mouse.

Si se usa e interpreta de la manera correcta, puede ser una mina de oro de datos y brindarle información increíble sobre el comportamiento del cliente y permitirle comprender mejor a sus clientes. El análisis de mapa de calor puede ayudarlo a detectar usuarios frustrados, brindar información para mejorar la optimización de su tasa de conversión, mejorar su embudo de conversión y brindar conocimiento personalizado del viaje del cliente a los equipos de productos.

Los mapas de calor se han utilizado de alguna forma desde finales de 1800, cuando Toussaint Loua usó un mapa de sombreado para visualizar los cambios demográficos sociales en todo París (Gehlenborg & Wong, 2012).

La tecnología de mapeo de calor por computadora fue registrada por primera vez a principios de la década de 1990 por el diseñador de software Cormac Kinney, quien creó una herramienta para mostrar gráficamente información del mercado financiero en tiempo real.

Se puede usar una variedad de esquemas de color al crear mapas de calor, incluida la escala de grises y el arcoíris. Sin embargo, a menudo se prefieren los mapas con esquemas de arco iris, ya que los humanos pueden percibir más tonos de color que de gris (aunque hay inconvenientes).

En términos generales, los colores más cálidos representan secciones "más usadas" o "más populares", mientras que los colores más fríos representan secciones del mapa que se usan con menos frecuencia (Rubkiewicz, 2022).

Sin embargo, cada mapa de calor funciona de manera diferente. Por ejemplo, los mapas de clic suelen utilizar diferentes tonos de un color: cuanto más oscuro es el color, más se hace clic en esa área.

Los mapas de calor son representaciones visuales de las reacciones de los usuarios en varias páginas de su sitio web, lo que proporciona un contexto visual para facilitar el análisis (Wilkinson & Friendly, 2009). Lo ayudan a recopilar información sobre el comportamiento de los visitantes, que luego puede usar para personalizar su sitio web para cumplir mejor con las expectativas de los visitantes, mejorando los embudos de conversión, aumentando las tasas de conversión, reduciendo las tasas de rebote o aumentando las ventas, entre otros objetivos.

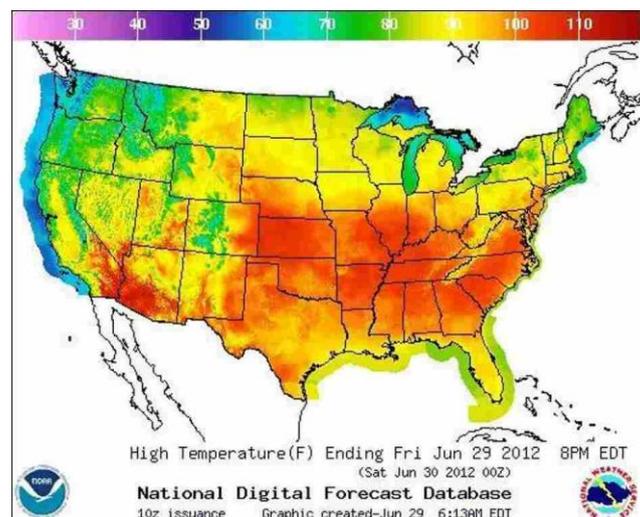


Figura 11 Ejemplo de un mapa de calor

Tomado de "Eye Tracking Heatmap: Front Row Seats To Your Visitor's Worldview", de Shanaz Khan, 2022

2.2.7 Público Objetivo

Un público objetivo es un grupo de personas identificadas como posibles clientes de una empresa. Las audiencias objetivo comparten rasgos demográficos similares que incluyen, entre otros (Scarton & Specia, 2018):

- Años
- Género
- ubicación
- Educación
- Estatus socioeconómico

Identificar al público objetivo como empresa puede ayudar a diseñar estrategias de marketing y definir a sus clientes principales. En lugar de gastar dinero y recursos tratando de atender a todos los consumidores, definir un público objetivo permite un alcance más intencional y personal a aquellos que tienen más probabilidades de comprar su producto o servicio.

La mejor manera de encontrar al público objetivo es pensar primero en las necesidades específicas que satisface un producto o servicio. Es importante identificar el punto de dolor y luego determinar quién tiene esos puntos de dolor. Por ejemplo, si se aprendió que crear sitios web es una molestia y comienza un negocio que diseña sitios web, el próximo paso sería averiguar quién necesita un sitio web que probablemente tenga problemas para crear uno. En este escenario, el público objetivo probablemente sea propietarios de pequeñas empresas.

Cuanto más específico pueda identificar la demografía de su público objetivo, más eficazmente se podrá anunciarles. En este ejemplo, sería mejor centrarse en los propietarios de pequeñas empresas que probablemente tengan menos conocimientos técnicos (Masco, 2008). En lugar de considerar a los propietarios de pequeñas empresas como su público objetivo, puede optar por centrarse en los propietarios de pequeñas empresas mayores de una edad específica en una ubicación particular.

Si un producto es de naturaleza muy general, no se tendrá que hacer tanta investigación de mercado para encontrar su público objetivo. Si la audiencia es

más específica, es importante recopilar datos sobre sus clientes para que pueda enfocarse más específicamente. Una forma de recopilar datos es ofrecer un precio especial o un código de cupón a quienes visitan su sitio web o negocio si completan una encuesta que captura la información que se necesita (Lake, 2019).

Las audiencias objetivo ayudan a las empresas a anunciarse de manera más eficiente, ya que sabe quién es su audiencia objetivo y cómo llegar a ella. Si bien es importante llegar a la mayor cantidad de personas posible y, a menudo, parece que centrarse solo en segmentos específicos de la población es una limitación, debe llegar directamente a los consumidores potenciales.



Figura 12 Infografía del público objetivo

Tomado de “Target audience infographic 10 option concept vector image”, de rashadashurov a través de Vector Stock

2.2.8 Tecnología de Datos

Antes de que se introdujeran las tecnologías de macrodatos, los datos se gestionaban mediante lenguajes de programación generales y lenguajes de consulta estructurados básicos. Sin embargo, estos lenguajes no fueron lo suficientemente eficientes para manejar los datos porque ha habido un crecimiento continuo en la información y los datos de cada organización y el dominio (Oussous et al., 2018). Es por eso que se volvió muy importante manejar datos tan grandes e introducir una tecnología eficiente y estable que atienda

todos los requerimientos y necesidades de los clientes y las grandes organizaciones, responsables de la producción y el control de los datos. Tecnologías de big data, la palabra de moda que se escucha mucho en los últimos tiempos para todas esas necesidades.

La tecnología de datos es un sector del mercado que desarrolla soluciones tecnológicas utilizadas en el análisis de big data y para controlar conjuntos de datos. Cada año, los consumidores generan muchos datos que pueden ser beneficiosos. Aquí es donde entra la tecnología. Ayuda a administrar los datos integrándolos desde diferentes fuentes, almacenándolos y organizándolos. Todo esto es bastante útil para analizar el comportamiento del consumidor y obtener información comercial invaluable para identificar nuevas estrategias y oportunidades de inversión.

Algunas Herramientas de Tecnología de Big Data son las siguientes (Pedamkar, 2019):

- Apache Spark
- NoSQL databases
- Apache Kafka
- Apache Oozie
- Apache Airflow
- Apache Beam
- ELK Stack
- Docker & Kubernetes
- TensorFlow
- Presto
- Polybase
- Hive

El volumen de datos creado está en continua expansión. Cada vez que se realiza una búsqueda, visita un sitio web, se compra en línea o incluso comenta una publicación, se contribuye a obtener más datos. Toda esta información puede ser muy valiosa si se extrae, procesa y utiliza correctamente. Imagine las decisiones que podría tomar su negocio si tuviera acceso a toda la información que se necesita.

La tecnología de datos es clave para utilizar los datos en sus diversas formas. Con la tecnología de datos, las empresas pueden (Zuiderwijk et al., 2015):

- Aumente la productividad
- Aumentar la consistencia organizacional
- Tome decisiones comerciales más perspicaces
- Identificar nuevas oportunidades de negocio

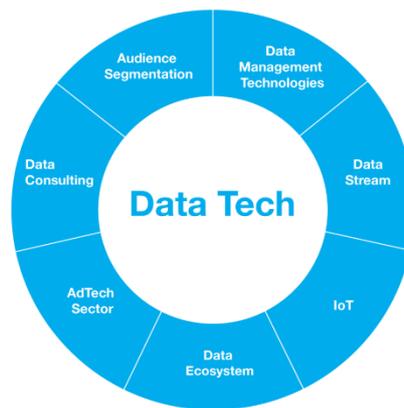


Figura 13 Composición de la tecnología de Datos

Tomado de "What is DataTech?", del equipo de OnAudience, 2019

2.2.9 Universidad Católica de Santiago de Guayaquil

La Universidad Católica de Santiago de Guayaquil es una institución de educación superior que tiene como principal objetivo formar profesionales socialmente responsables basados en el estudio, conservación, promoción y difusión de la ciencia y la cultura, enfatizando sus valores originales para garantizar el óptimo desarrollo y promoción de los ecuatorianos. a la convivencia, a la justicia social, creativa en el marco del respeto y promoción de la paz, los valores y los derechos humanos, brindando institucionalmente un sello auténtico de la inspiración cristiana y del mensaje de Cristo, como lo determina la Iglesia Católica, promovida por la comunidad y actividad trascendental, consciente de su papel social en la solución de los problemas sociales contemporáneos en el mundo universitario, cumpliendo su misión académica como instituto, educación y cultura abierta a todas las corrientes generales de pensamiento universal.

Fue creada el 17 de mayo de 1962, por lo que actualmente cuenta con 60 años de vida institucional (Universidades de Ecuador, s/f).

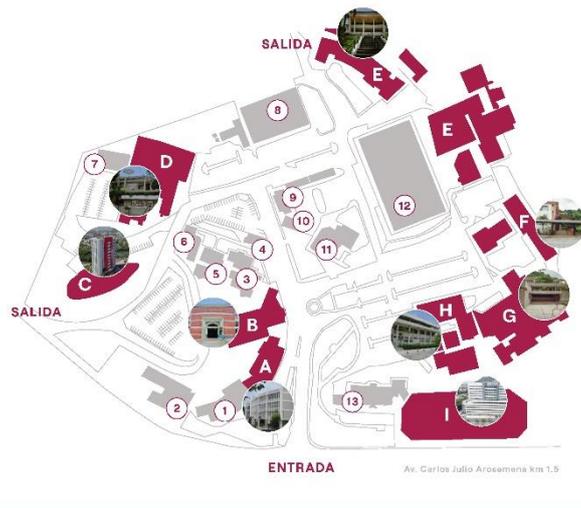


Figura 14 Mapa de la UCSG

Tomado de la página oficial de la UCSG

La Universidad está ubicada en Ecuador, en la provincia del Guayas, ciudad de Guayaquil, en el Km 1.5 de la Av. Carlos Julio Arosemena.

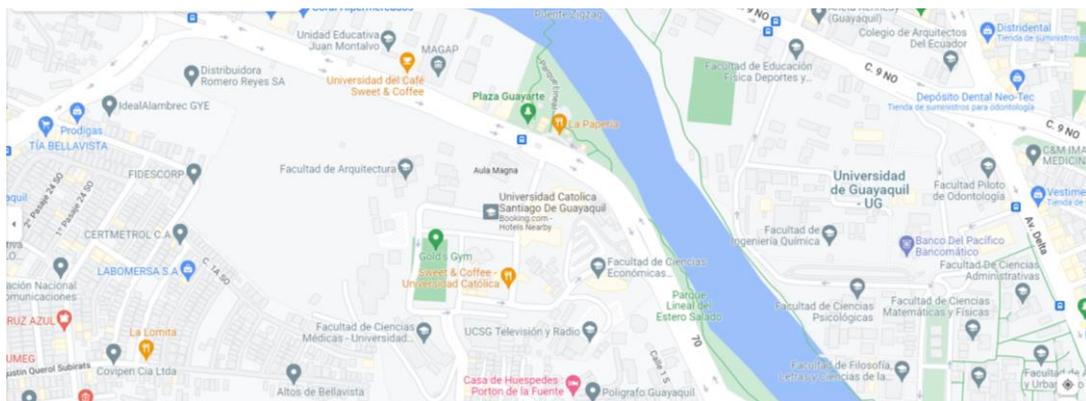


Figura 15 Ubicación de la UCSG en la ciudad de Guayaquil

Tomado de Google Maps, Google

La Universidad Católica de Santiago de Guayaquil se encuentra hoy en día en un puesto más bajo al 1401 del Ranking QS de las mejores Universidades del mundo, y entre el puesto 251 y 300 del Ranking de las mejores universidades Latinoamericanas (TopUniversities, s/f).

Rank	University	Overall Score
1401+	 Universidad Autónoma de Guadalajara (UAG) <small>Guadalajara, Mexico</small>	
1401+	 Universidad Católica Boliviana "San Pablo" <small>La Paz, Bolivia</small>	
1401+	 PUniversidad Católica de Santiago de Guayaquil <small>Guayaquil, Ecuador</small>	
1401+	 Universidad Católica de Temuco <small>Temuco, Chile</small>	
1401+	 Universidad Mayor de San Simón <small>Cochabamba, Bolivia</small>	
1401+	 Universidad de San Nicolás de Hidalgo <small>Morelia, Mexico</small>	

Figura 16 Ubicación de UCSG en Ranking QS

Tomado de “QS World University Rankings 2023: Top global universities” de TopUniversities

Rank	University	Overall Score
251-300	 Universidad Autónoma de Chihuahua (UACH) <small>Chihuahua, Mexico</small>	
251-300	 Universidad Autónoma del Caribe <small>Barranquilla, Colombia</small>	
251-300	 PUniversidad Católica de Santiago de Guayaquil <small>Guayaquil, Ecuador</small>	
251-300	 Universidad Centroamericana "José Simeón Cañas" <small>San Salvador, El Salvador</small>	
251-300	 Universidad Científica del Sur <small>Lima, Peru</small>	
251-300	 Universidad de Atacama <small>Copiapó, Chile</small>	

Figura 17 Ubicación de UCSG en Ranking de Latam

Tomado de “QS World University Rankings 2023: Top global universities” de TopUniversities

Actualmente, La UCSG posee 28 carreras universitarias, 14 posgrados y 28 carreras cortas. Entre las carreras, se puede estudiar alguna de las 23 licenciaturas y 5 ingenierías.

La Universidad Católica de Santiago de Guayaquil es además una universidad a distancia. Hoy en día dicta dicta 10 carreras a distancia, entre ellas 3 licenciaturas a distancia y 7 posgrados a distancia (7 diplomaturas a distancia) (Universidades de Ecuador, s/f).

<p>ARQUITECTURA Y DISEÑO</p> <p>Arquitectura P</p> <p>Diseño de Interiores P</p> <p>Diseño Gráfico P</p>	<p>FAH FACULTAD DE ARTES Y HUMANIDADES</p> <p>Animación Digital P</p> <p>Artes Musicales P</p> <p>Cine P</p> <p>Pedagogía de los Idiomas Nacionales y Extranjeros – Inglés P</p>	<p>JSP JURISPRUDENCIA Y CIENCIAS SOCIALES Y POLÍTICAS</p> <p>Derecho P</p> <p>Trabajo Social P</p>
<p>CM CIENCIAS MÉDICAS</p> <p>Enfermería P</p> <p>Medicina P</p> <p>Nutrición y Dietética P</p> <p>Odonatología P</p> <p>Fisioterapia P</p>	<p>E+D EDUCACIÓN TÉCNICA PARA EL DESARROLLO</p> <p>Agroindustria P</p> <p>Agropecuaria P</p> <p>Electricidad P</p> <p>Electrónica y Automatización P</p> <p>Medicina Veterinaria P</p> <p>Telecomunicaciones P</p>	<p>EE Facultad de Economía y Empresa</p> <p>Administración de Empresas P</p> <p>Comercio P</p> <p>Comercio Exterior P</p> <p>Contabilidad y Auditoría P</p> <p>Economía P</p> <p>Emprendimiento e Innovación Social P</p> <p>Mercadotecnia P</p> <p>Negocios Internacionales P</p> <p>Turismo P</p>
<p>CC CIENCIAS DE LA COMUNICACIÓN, LETRAS Y LINGÜÍSTICA</p> <p>Comunicación P</p> <p>Educación P</p> <p>Literatura P</p> <p>Psicología Clínica P</p> <p>Psicología Organizacional P</p>	<p>ING INGENIERÍA</p> <p>Computación P</p> <p>Ingeniería Civil P</p>	

Figura 18 Oferta académica de Grado de la UCSG

Tomado de la página oficial de la UCSG

Actualmente la Universidad Católica de Santiago de Guayaquil posee un cercano a 11500 alumnos en Grado y 1000 en posgrado, y un cercano de 820 docentes en todas las facultades disponibles.

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1 Tipo de investigación

La investigación utilizada en el presente proyecto es del tipo descriptiva. La investigación descriptiva se refiere a los métodos que describen las características de las variables en estudio. Esta metodología se enfoca en responder preguntas relacionadas con el “qué” más que con el “por qué” del sujeto de investigación. El objetivo principal de esta investigación es simplemente describir la naturaleza del suplente demográfico en lugar de centrarse en el "por qué".

Se denomina método de investigación observacional ya que ninguna de las variables del estudio se ve influenciada durante el proceso de investigación (Koh & Owen, 2000).

Por ejemplo, supongamos que una marca con sede en el Reino Unido está tratando de establecerse en Nueva York y quiere comprender la demografía de los compradores que generalmente compran a marcas similares.

En la investigación descriptiva, la información recopilada de la encuesta se centrará únicamente en la demografía de la población. Descubrirá detalles sobre los patrones de compra de diferentes cohortes de edad en Nueva York. No estudiará por qué existen tales patrones, porque la marca está tratando de establecerse en Nueva York.

Todo lo que quieren entender es el comportamiento de compra de la población, y no por qué existen tales asociaciones. Es una parte de la investigación de mercado cuantitativa o estudio de investigación social que implica realizar una investigación de encuesta utilizando variables cuantitativas en una herramienta de investigación de mercado o herramienta de investigación social (Grimes & Schulz, 2002).

La investigación descriptiva es una opción adecuada cuando el objetivo de la investigación es identificar características, frecuencias, tendencias y categorías.

Es útil cuando aún no se sabe mucho sobre el tema o problema. Antes de que pueda investigar por qué sucede algo, debe comprender cómo, cuándo y dónde sucede.

La investigación descriptiva generalmente se define como un tipo de investigación cuantitativa, aunque la investigación cualitativa también se puede utilizar con fines descriptivos. El diseño de la investigación debe desarrollarse cuidadosamente para garantizar que los resultados sean válidos y confiables.

Encuestas

La investigación de encuestas le permite recopilar grandes volúmenes de datos que se pueden analizar en busca de frecuencias, promedios y patrones. Los usos comunes de las encuestas incluyen (Svensson, 1984):

- Describir la demografía de un país o región.
- Medir la opinión pública sobre temas políticos y sociales.
- Evaluar la satisfacción con los productos de una empresa o los servicios de una organización

Observaciones

Las observaciones le permiten recopilar datos sobre comportamientos y fenómenos sin tener que depender de la honestidad y precisión de los encuestados. Los investigadores psicológicos, sociales y de mercado suelen utilizar este método para comprender cómo actúan las personas en situaciones de la vida real.

La observación de entidades y fenómenos físicos también es una parte importante de la investigación en las ciencias naturales. Antes de que pueda desarrollar hipótesis, modelos o teorías comprobables, es necesario observar y describir sistemáticamente el tema que se investiga.

Estudios de caso

Un estudio de caso se puede utilizar para describir las características de un tema específico (como una persona, grupo, evento u organización). En lugar de recopilar un gran volumen de datos para identificar patrones a lo largo del

tiempo o la ubicación, los estudios de casos recopilan datos detallados para identificar las características de un tema definido de manera limitada.

En lugar de tratar de describir hechos generalizables, los estudios de casos a menudo se enfocan en casos inusuales o interesantes que desafían las suposiciones, agregan complejidad o revelan algo nuevo sobre un problema de investigación.

Entre las muchas, las siguientes son las principales características de este tipo de investigación (Nassaji, 2015):

1. Investigación cuantitativa:

Es de naturaleza cuantitativa ya que intenta recopilar información y analizarla estadísticamente. Este tipo de investigación es una poderosa herramienta de investigación que permite a un investigador recopilar datos y describir la demografía de los mismos con la ayuda del análisis estadístico. Por lo tanto, es un método de investigación cuantitativa.

2. Naturaleza de las variables:

Las variables incluidas en esta investigación no están controladas. No se manipulan de ninguna manera. La investigación descriptiva utiliza principalmente métodos de observación y, por lo tanto, el investigador no puede controlar la naturaleza y el comportamiento de las variables en estudio.

3. Estudios transversales:

En este tipo de investigación se estudian diferentes secciones de un mismo grupo. Por ejemplo, para estudiar las preferencias de moda de Nueva York, el investigador puede estudiar tanto a la Generación Z como a los Millennials de la misma población en Nueva York.

4. Dirige la investigación futura:

Dado que esta investigación señala los patrones entre las variables y los describe, los investigadores pueden estudiar más a fondo los datos recopilados aquí. Guía a los investigadores para que averigüen más por qué se han encontrado tales patrones y la asociación entre ellos. Por lo tanto, brinda a los investigadores una dirección hacia una investigación de mercado perspicaz.

Habiendo puesto de manifiesto todo lo anterior, se llega sin ninguna duda a la conclusión de que el método de investigación descriptivo es el ideal para continuar hacia adelante con todo lo que el presente proyecto requiere.

3.2 Diseño de la investigación

En el desarrollo del proyecto actual se realizará un levantamiento de información a una persona que este al tanto de las necesidades de información con respecto a los estudiantes de la facultad de ingeniería de la Universidad Católica de Santiago de Guayaquil; y que también conozca acerca de las decisiones que se toman de manera directa e indirecta en las cuales la ubicación geográfica de los estudiantes tenga influencia.

Aquella persona responderá a cuestiones referentes a las necesidades informativas de los estudiantes, y dará su opinión acerca de la manera en que se presentará la información, la cual será tomada muy en cuenta.

El objetivo final de la consulta es dejar establecido desde el comienzo si existe algún tipo de deficiencia en la información geográfica disponible acerca de los alumnos de la Facultad de Ingeniería, y en el caso de que sea de aquella manera, hacer todo lo necesario para enmendar aquello. También se afinarán detalles referentes a la forma de presentación de los datos geográficos de los estudiantes.

La aplicación que será utilizada en la programación de la herramienta debe contener las siguientes características:

- Debe trabajar con una API de Google Maps
- Debe tener una librería para un mapa de calor espacial
- Debe ser una herramienta de desarrollo de analítica de datos (Python)

3.3 Técnicas e instrumentos para obtención de información

La herramienta de levantamiento de información que fue escogida para el presente proyecto es la entrevista. La misma estará dividida en dos temas principales, cada uno con sus respectivos subtemas.

Cabe recalcar que la entrevista a utilizar es el tipo semiestructurada, ya que esta se caracteriza por carecer de preguntas específicas predefinidas, por el contrario, lo que si está presente son temas que se tratan. La misma será realizada a una única persona.

La razón por la cual la entrevista fue escogida como la herramienta para obtener todos los datos necesarios es que es necesario conocer acerca de varios aspectos como: 1) Necesidades de información de los estudiantes de la Facultad de Ingeniería. 2) Forma de presentar la información a través de la herramienta.

Como se van a tratar temas, es de esperar que las respuestas obtenidas sean abiertas, lo cual aumentará en gran medida la facilidad con la que se pueden analizar y también permitirá revisarlos más en profundidad, así como presentarlos de una manera mucho más eficiente.

Anteriormente se mencionó (ver sección 3.2) que la persona a consultar debe tener presentes las necesidades de información de los estudiantes y de las decisiones que se toman en la universidad con influencia de la ubicación geográfica de estos. La persona escogida para obtener los datos es la Ing. Ana Camacho, directora de la Carrera de Ingeniería en Ciencias de la Computación.

La consulta que realizó tiene que ver con cómo se deberían utilizar los datos extra que se obtengan de los estudiantes.

3.4 Análisis de resultados

La persona entrevistada indicó lo siguiente:

Le pareció que toda la información extra que se obtenga sea también visualizada de una manera prudente. Información como el género de la persona, el colegio del cual proviene, si es que es oriundo de dentro o fuera de Guayaquil. También contempló la posibilidad de que ya conociendo a través del mapa de calor los sectores con mayor demografía se pueda hacer una campaña promocional de admisión. También recalcó la importancia que tendría esto al momento de ver las practicas preprofesionales de los estudiantes y la movilidad que deben hacer para realizarlas, así como el tramite para asignar el lugar. También se mostró de acuerdo en que la información extra se muestre luego de hacer clic en un determinado sector. Además, mostró su deseo de ver gráficas

en lo referente a la densidad demográfica. Y por último vio mucho potencial el proyecto para en el futuro ayudara determinar rutas de buses de la UCSG teniendo en cuenta los datos demográficos del mapa de calor.

En la sección 3.2 se expusieron los requerimientos necesarios que debe tener la aplicación con la cual se va a programar. Se ha llegado a la conclusión de que la herramienta que cumple con todo aquello es Python, un lenguaje de programación de código abierto.

Actualmente el mismo cuenta con muchos competidores para el desarrollo web, sin embargo, hay muchas razones por las cuales Python es mucho mejor.

A continuación, se muestran datos comparativos este lenguaje frente a otros disponibles.

Python vs Java

Point of Difference	Python	Java
Program run time	Python program runs slower than Java	Java program runs faster than python
Program development time	Takes less time than Java	Takes more time for developing the same program
Length of the code	Python code is 3-5 times shorter than Java code	Java code is longer than Python
Type	Python is dynamically typed	Java is not dynamically typed

Tabla 1 Comparación entre Python y Java

Tomado de “How Python Is Different From Other Languages” de Ajay Malik, 2020

Python vs JavaScript

Point of Difference	Python	JavaScript
---------------------	--------	------------

Object-based	A python is an object-based programming language	JavaScript is also an object-based scripting language
Type	Python can be used as a scripting language as well as a programming language	JavaScript is only used for the scripting language

Tabla 2 Comparación entre Python y JavaScript

Tomado de “How Python Is Different From Other Languages” de Ajay Malik, 2020

Python vs Perl

Point of Difference	Python	Perl
Basic	Python is very close to Perl	Perl is also very close to Python programming language
Programming methodologies	Python supports common programming methodologies such as data structure and object-oriented programming	Perl supports common application-oriented tasks like built-in regular expressions, file scanning and reports generating features

Tabla 3 Comparación entre Python y Perl

Tomado de “How Python Is Different From Other Languages” de Ajay Malik, 2020

Python vs Tcl

Point of Difference	Python	Tcl
---------------------	--------	-----

Typed	Python is a general-purpose language programming language	Tcl is application extension language and stand-alone programming language
Data type	Python has a rich Datatype set for storing values	Tcl stores all type values in a string mean TCL is weak in the data structure
Program execution time	Python program executes faster	Tcl program executes slower than python program
Modularity	Python supports module for writing large programs in a small module	Tcl also has lack of module for writing a large program

Tabla 4 Comparación entre Python y TCL

Tomado de “How Python Is Different From Other Languages” de Ajay Malik, 2020

Python vs Smalltalk

Point of Difference	Python	Smalltalk
Typed	Python is a dynamically typed language	Smalltalk is not dynamically typed programming
Standard library	Python's library has more facilities for dealing with the internet and WWW realities such as email, HTML and FTP	Smalltalk's standard library of collection data types is more refined

Tabla 5 Comparación entre Python y Smalltalk

Tomado de “How Python Is Different From Other Languages” de Ajay Malik, 2020

Python vs C++

Point of Difference	Python	C++
Length of code	Python code is 5-10 time shorter than C++ code	C++ code is 5-10 times longer than Python code
OOPS	Python is pure OOPS programming language	C++ is not a pure OOPS programming language

Tabla 6 Comparación entre Python y C++

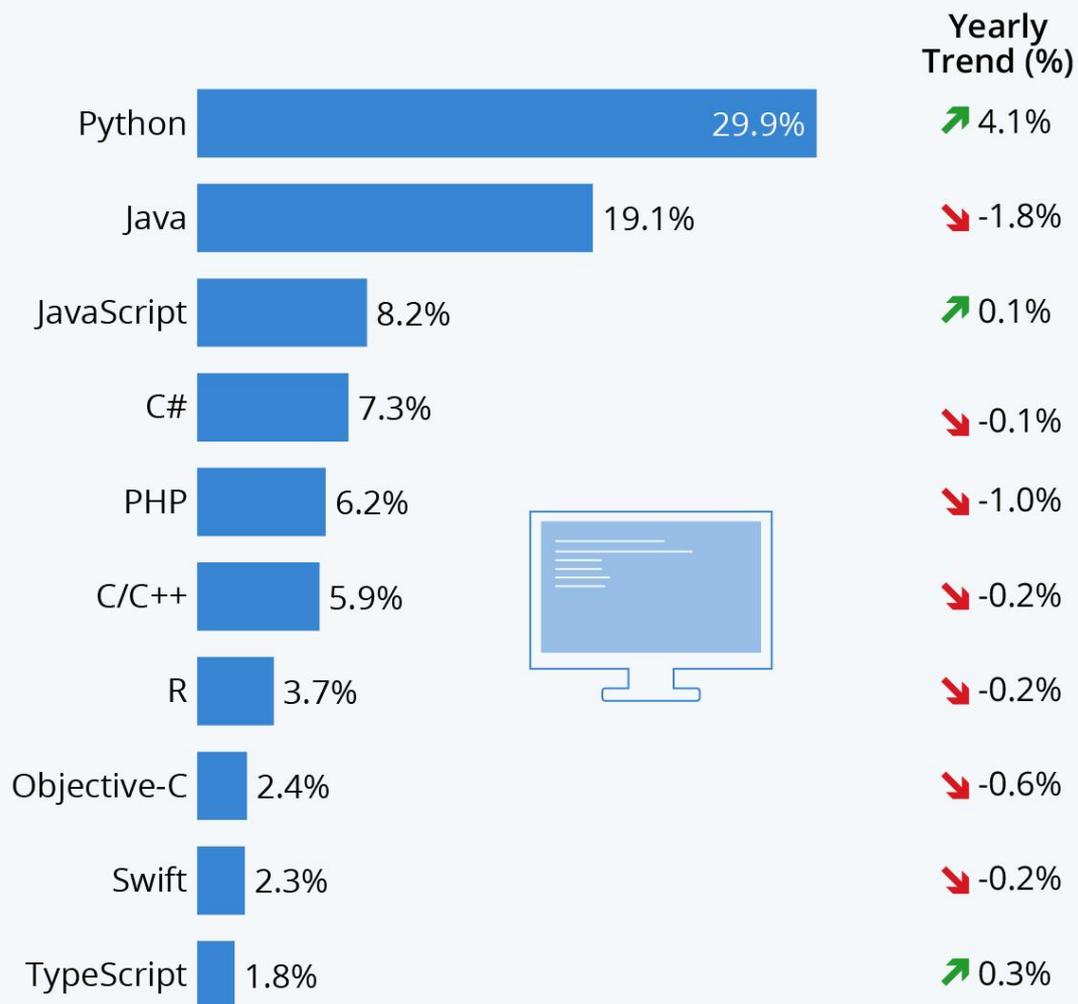
Tomado de "How Python Is Different From Other Languages" de Ajay Malik, 2020

	 Python	 JavaScript (JS)	 C++	 PHP	 Go
Learning Curve	Quite easy	Comparatively harder	Hard	Quite easy	Hard
Cross-Platform	Yes	Yes	Yes	Yes	Yes
Compatibility with operating systems	Windows, macOS, and Linux	Windows, macOS, and Linux	Mac, Windows, Linux, BSD and more	Windows, Linux	Mac, Windows, Linux, BSD and more
Nature	Dynamically typed	Dynamically typed	Statically typed	Dynamically typed	Statically typed
Performance	Comparatively Slower	Fast	Fast	Comparatively Slower	Fast
Type	Interpreted	Interpreted	Compiled	Interpreted	Compiled
Libraries	Yes	Not many	Not many	Not many	Not many

Figura 19. Comparativa de Python con otros lenguajes en varios aspectos

Python Remains Most Popular Programming Language

Popularity of each programming language based on share of tutorial searches in Google



Yearly trend compares percent change from Feb 2019 to Feb 2020
Sources: GitHub, Google Trends



Figura 20. Popularidad de Python frente a otros lenguajes

Tomado de “Top Reasons Why Python Development is Widely Used in Enterprise” del equipo de ICORE, 2021

Language Rank	Types	Spectrum Ranking
1. Python	  	100.0
2. C++	  	99.7
3. Java	  	97.5
4. C	  	96.7
5. C#	  	89.4
6. PHP		84.9
7. R		82.9
8. JavaScript	 	82.6
9. Go	 	76.4
10. Assembly		74.1

Figura 21. Rango de Python frente a otros lenguajes

Tomado de “How is Python Different from Other Programming Languages?” de Michael Garbade, 2018

CAPÍTULO IV

PROPUESTA TECNOLÓGICA

En esta sección se busca hacer un análisis a detalle del proceso de implementación de la herramienta. Se revisará tres cosas principalmente: las herramientas utilizadas, las librerías de Python que fueron de ayuda y los tipos de datos presentes.

4.1 Tratamiento de datos

Para este desarrollo el tratamiento de los datos es la pieza clave de este. Sin embargo, esto no quiere decir que sea un proceso largo ni mucho menos complicado. Es más, se trata de ser lo más específicos posibles; con lo cual la modificación de datos se ajusta al formato que nos pide la herramienta que hará uso de ellos. En este caso se cambió algunas abreviaturas por palabras completas; además de utilizar una columna de los datos para denotar el cantón de residencia. Por último, agregar al final de la cadena la palabra Ecuador. Con este tratamiento todas las direcciones se van a mostrar en el mapa de calor.

Se comienza por agregar el archivo que se usará para realizar el tratamiento, este puede ser cualquier archivo siempre y cuando se pueda convertir a un DataFrame de la librería Pandas, se ingresan las columnas de interés y se declaran variables que almacenan las direcciones vacías, direcciones no procesadas y localizaciones, esta última almacenará las direcciones luego del tratamiento e obtención de latitud y longitud a partir de la dirección.

```
data = pd.read_excel(r'Archivo a tratar.extension') #funcion lee excel
df = pd.DataFrame(data, columns=['DIRECCION', 'CANTON_RESIDENCIA',
                                'CARRERA', 'SEXO', 'DISCAPACIDAD', 'SNNA']) #columnas de interés
direcciones_vacias = 0 #direcciones vacias iniciadas en 0
direcciones_no_procesadas = [] #direcciones no procesadas
localizaciones = [] #direcciones finales luego del tratamiento
```

Luego se ingresa la API Key para el uso de la API de Google Maps, y se da el nombre por defecto a la aplicación de Dash.

```
API_KEY = 'AIzaSyANGrZfSgQ_1L55qfMBWojglsOacfkIuwo'
```

```
gmaps = googlemaps.Client(API_KEY)

app = Dash(__name__)
```

Se codificó en total 2 funciones que limpiarán y una última que retornará la latitud y longitud de la dirección.

limpia recibe como único parámetro la dirección, se realiza un split a la dirección para retornar su secuencia de palabras y usar esa secuencia en una sentencia iterativa, se quitan las abreviaturas como “CDLA” a “CIUDADELA”, se unen las palabras y se retorna la cadena habiendo limpiado las abreviaturas, además de agregar al final la palabra “ECUADOR”.

identidad recibe como único parámetro la dirección, la función solo devolverá verdadero mientras no sea “ ” o sea la palabra ‘DOMICILIO’, en caso de serlo se sumará 1 a la variable de direcciones vacias.

agregaDir recibe como único parámetro la dirección, aquí se hace uso de la API de google maps, usamos la función de la API que nos retorna todos los datos de la dirección que le mandamos, sin embargo, solo retornaremos latitud y longitud, luego de [0] (para un retorno sin []) tenemos que ir a atributo ‘geometry’ y atributo ‘location’ para obtenerlas, siempre y cuando la API nos retorne los datos, es decir, que se haya podido procesar, de ser este el caso se hará un append de esa dirección a la variable de direcciones no procesadas y no retornará nada.

```
def limpia(dir):
    aux = str(dir).split()
    for index in range(len(aux)):
        if(aux[index].upper() == 'CDLA'):
            aux[index] = 'CIUDADELA'
        if(aux[index].upper() == 'URB'):
            aux[index] = 'URBANIZACIÓN'
        if(aux[index].upper() == 'COOP'):
            aux[index] = 'COOPERATIVA'
        if(aux[index].upper() == 'V'):
            aux[index] = 'VILLA'
        if(aux[index].upper() == 'AV'):
            aux[index] = 'AVENIDA'
    aux = ' '.join(aux)
```

```

return aux + ' ECUADOR'

def identidad(x):
    if(x == '' or str(x).upper() == 'DOMICILIO'):
        global direcciones_vacias
        direcciones_vacias += 1
        return False
    return True

def agregaDir(dir):
    global direcciones_no_procesadas
    geocode_result = gmaps.geocode(dir)
    if(geocode_result != []):
        return geocode_result[0]['geometry']['location']
    direcciones_no_procesadas.append(dir)

```

Se agregará una columna al DataFrame llamada 'dir_c' en esta se concatenará la dirección y el cantón de residencia (contiene los cantones del Ecuador) además de declarar una variable que almacene el DataFrame luego de limpiar los datos "nan" (Not a Number) es decir, datos que no se hayan leído correctamente, y se reemplazan con "".

```

df['dirs_c'] = df['DIRECCION'] + ' ' + df['CANTON_RESIDENCIA']
cleanData = df.fillna('')

```

Para terminar el proceso de tratamiento de los datos, se va a iterar la variable cleanData que contiene los datos reemplazados con "", verificamos si la dirección no es "" o 'DOMICILIO' con la función identidad, en caso de retornar True se usará la función limpia en la columna de 'dirs_c' y luego la función agregaDir en la nueva cadena, y se asignará a la variable geocode_result, en caso de que la función identidad retorne False, se asignará un valor de None a geocode_result, luego verificamos si geocode_result es diferente de none, además de verificar límites de latitudes y longitudes para que se muestren dentro del Ecuador, de estar fuera, no se graficarán y se agregarán a las direcciones no procesadas, de estar dentro, se agregarán columnas de interés a geocode_result, y se hará un append de la misma a la variable de localizaciones. Luego se creará el DataFrame con las localizaciones finales, un pprint para observar por consola las cabeceras y cola de las direcciones, y agregar la columna 'z' para que se vuelvan ubicables en el mapa. Se ubica con centro en

latitud -2, longitud -79 y zoom en 5 para observar todo el Ecuador y Galápagos, un radio de 8 para observar unos puntos no tan pequeños ni muy grandes, y mostrar en los Tooltips (hover_data) las columnas de interés. Se realiza un pprint final por consola con la cantidad de direcciones vacias y no procesadas.

```
for index, row in cleanData.iterrows():
    esDir = identidad(row['DIRECCION'])
    if(esDir == True):
        geocode_result = agregaDir(limpia(row['dirs_c']))
    else:
        geocode_result = None

    if(geocode_result != None):
        if((geocode_result['lat'] > -4 and geocode_result['lng'] < -65)
and (geocode_result['lat'] < 2 and geocode_result['lng'] > -99)):
            geocode_result['direccion'] = row['DIRECCION']
            geocode_result['carrera'] = row['CARRERA']
            geocode_result['sexo'] = row['SEXO']
            geocode_result['discapacidad'] = row['DISCAPACIDAD']
            geocode_result['sna'] = row['SNA']
            localizaciones.append(geocode_result)
        else:
            direcciones_no_procesadas.append(row['DIRECCION'])

localizacionesDF = pd.DataFrame(localizaciones)
pprint(localizacionesDF)
localizacionesDF['z'] = 1

fig = px.density_mapbox(localizacionesDF, lat='lat', lon='lng', z='z',
radius=8,
                        center=dict(lat=-2, lon=-79), zoom=5, hover_data
= {'lat' : False,
   'lng' : False,
   'z' : False,
   'direccion' : True,
   'carrera' : True,
   'sexo' : True,
   'discapacidad' : True,
   'sna' : True},
                        mapbox_style="stamen-terrain")
```

```
pprint('Direcciones Vacias: ' + str(direcciones_vacias))
pprint('Direcciones No Procesadas: ' +
str(len(direcciones_no_procesadas)))
```

```

55
56
57 df['dirs_c'] = df['DIRECCION'] + ' ' + df['CANTON_RESIDENCIA']
58 cleanData = df.fillna("")
59
60 for index, row in cleanData.iterrows():
61     esDir = Identity(row['DIRECCION'])
62     if(esDir == True):
63         geocode_result = agrapair(limpia(row['dirs_c']))
64     else:
65         geocode_result = None
66
67     if(geocode_result != None):
68         (variable) geocode_result: Any
69         if((geocode_result['lat'] > -4 and geocode_result['lng'] < -65 and geocode_result['lat'] < 2 and geocode_result['lng'] > -99)):
70             geocode_result['direccion'] = row['DIRECCION']
71             geocode_result['carrera'] = row['CARRERA']
72             geocode_result['sexo'] = row['SEXO']
73             geocode_result['discapacidades'] = row['DISCAPACIDAD']
74             geocode_result['sema'] = row['SEMA']
75             locations.append(geocode_result)
76         else:
77             direcciones_no_procesadas.append(row['DIRECCION'])

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
 Python 3.11.0 64-bit
 Activar Windows
 Ve a Configuración para activar Windows.

Figura 22 Tratado de Datos 1

```

18 app = Dash(__name__)
19
20 def getCarrera(dirs, df):
21
22     def limpia(dir):
23         aux = str(dir).split()
24         for index in range(len(aux)):
25             if(aux[index].upper() == "CARRERA"):
26                 aux[index] = "CARRERA"
27             elif(aux[index].upper() == "URBANIZACION"):
28                 aux[index] = "URBANIZACION"
29             elif(aux[index].upper() == "COP"):
30                 aux[index] = "COP"
31             elif(aux[index].upper() == "CORREDORES"):
32                 aux[index] = "CORREDORES"
33             elif(aux[index].upper() == "VIA"):
34                 aux[index] = "VIA"
35             elif(aux[index].upper() == "AV"):
36                 aux[index] = "AV"
37             elif(aux[index].upper() == "AVENIDA"):
38                 aux[index] = "AVENIDA"
39         aux = " ".join(aux)
40         return aux + " " + CODOR
41
42     def identity(x):
43         if(x == "" or str(x).upper() == "DIRECCION"):

```

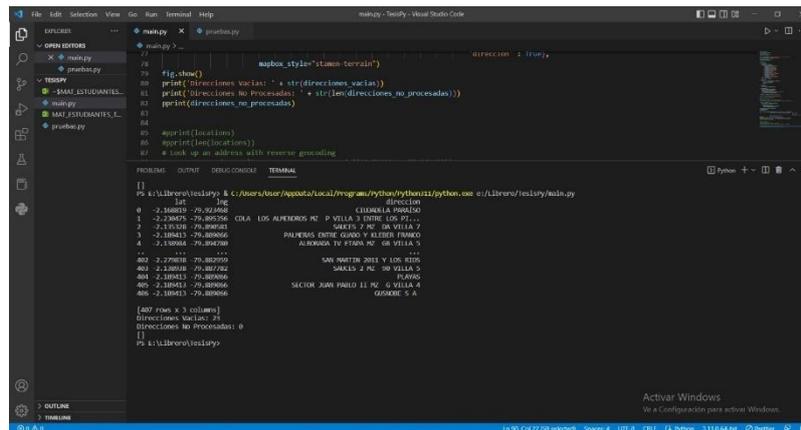
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
 Python 3.11.0 64-bit
 Activar Windows
 Ve a Configuración para activar Windows.

Figura 23 Tratado de Datos 2

4.2 Herramientas utilizadas en el Desarrollo

4.2.1 Python

Para el desarrollo y creación de toda la codificación pertinente era mas que necesario trabajar con un lenguaje de programación. Y como se mencionó con anterioridad, el escogido fue Python. Python es un lenguaje de programación de propósito general y alto nivel. Su filosofía de diseño enfatiza la legibilidad del código con el uso de sangría significativa. Python se utilizó para programar todo relacionado con las funcionalidades, presentación y el procesamiento de los datos. La versión del lenguaje utilizada es la 3.11.0.



```
def main():
    # Crear un objeto de direcciones
    direcciones = Direcciones()

    # Agregar direcciones
    direcciones.agregar('Calle 123', 123, 456)
    direcciones.agregar('Calle 456', 456, 789)
    direcciones.agregar('Calle 789', 789, 101)

    # Mostrar direcciones
    direcciones.mostrar()

    # Filtrar direcciones por longitud
    direcciones.filtrar(5)

    # Ordenar direcciones por longitud
    direcciones.ordenar()

    # Buscar direcciones por nombre
    direcciones.buscar('Calle 123')

    # Calcular estadísticas
    direcciones.calcular_estadisticas()

if __name__ == '__main__':
    main()
```

```
PS C:\Libreria\libreria> python main.py
[[]]
lat
lon
direccion
0 -2.348805 -79.263458 COLUMELA PARQUE
1 -2.329075 -79.260350 COLA LOS ALBERDOS PT P VILLA 3 TRONC LOS 92
2 -2.315108 -79.260281 SAKES 7 PM DA VILLA 7
3 -2.318411 -79.260066 PALMEROS ENTRE GORDO Y ALBERTO FERRAZ
4 -2.318868 -79.262738 ALBERGADA TV ESTAD PT 08 VILLA 5
5 -2.318868 -79.262738
602 -2.278818 -79.262759 SAN MARTIN 2011 V LOS RIOS
802 -2.280818 -79.262752 SAKES 2 PM 30 VILLA 3
804 -2.318411 -79.260066 FLAVES
805 -2.318411 -79.260066 SECTOR JUAN PABLO LE RO 0 VILLA 4
806 -2.318411 -79.260066 GORDON 5 A
[607 rows x 3 columns]
Direcciones vacías: 28
Direcciones no procesadas: 0
[[]]
PS C:\Libreria\libreria>
```

Figura 24 Interfaz de Visual Studio con código de Python

4.2.2 Google maps

Era necesario trabajar de manera directa con un programa de Sistema de posicionamiento global, y Google Maps cumplía con ese requerimiento y con muchos más que podrían facilitar el Desarrollo del trabajo. Google Maps es un servidor de aplicaciones de mapas en la web que pertenece a Alphabet Inc. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo e incluso la ruta entre diferentes ubicaciones o imágenes a pie de calle con Google Street View. La función que cumplió Google Maps en el Desarrollo fue la de traducir las direcciones que se le enviaba y convertirlos en datos de latitud y longitud.

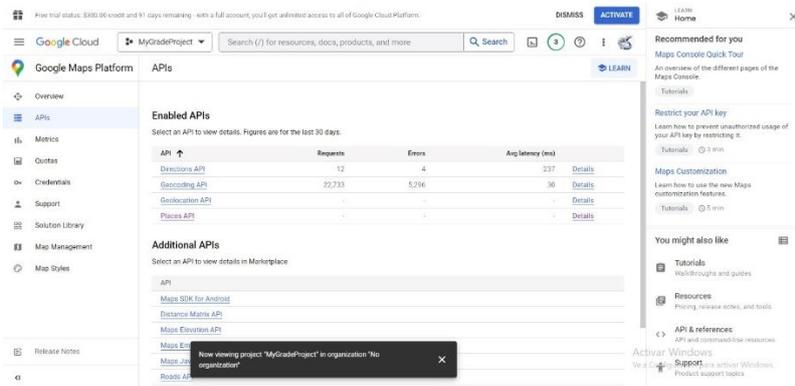


Figura 25 Plataforma de Google Maps

4.3 Librerías de Python utilizadas para la programación

4.3.1 Google maps

En python existe una librería que nos permite establecer el contacto requerido con Google Maps. Esta librería lleva el mismo nombre que el sistema de geolocalización de Google. El cliente de Python para Google Maps Services es una biblioteca de cliente de Python para las siguientes API de Google Maps:

- API de indicaciones
- API de matriz de distancia
- API de elevación
- API de geocodificación
- API de geolocalización
- API de zona horaria
- API de carreteras
- API de lugares
- API estática de mapas
- API de validación de direcciones

De las APIs mencionadas con anterioridad las que se utilizaron fueron las de geocodificación y geolocalización.


```

File Edit Selection View Go Run Terminal Help
main.py - Testify - Visual Studio Code
EXPLORER
main.py X
pruebas.py
main.py
pruebas.py
TESSPY
SMAT ESTUDIANTES...
main.py
MAT_ESTUDIANTES_I...
pruebas.py
main.py
pruebas.py
64
65 locationsDF = pd.DataFrame(locations)
66 pprint(locationsDF)
67 locationsDF['z'] = 1
68
69 fig = px.density_mapbox(locationsDF, lat='lat', lon='lon', z='z', radius=10,
70                        center=dict(lat=0, lon=100), zoom=0, hover_data = {
71                            'lat': True,
72                            'lon': True,
73                            'z': False,
74                            'direccion': True},
75                        mapbox_style='stamen-terrain')
76
77 fig.show()
78 print('Direcciones vacias: ' + str(direcciones_vacias))
79 print('Direcciones no procesadas: ' + str(len(direcciones_no_procesadas)))
80 pprint(direcciones_no_procesadas)
81
82 pprint(locations)
83 pprint(len(locations))
84 # Look up an address with reverse geocoding
85 #reverse_geocode_result = gmaps.reverse_geocode((40.714224, -73.061452))
86 #print(reverse_geocode_result)
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
```

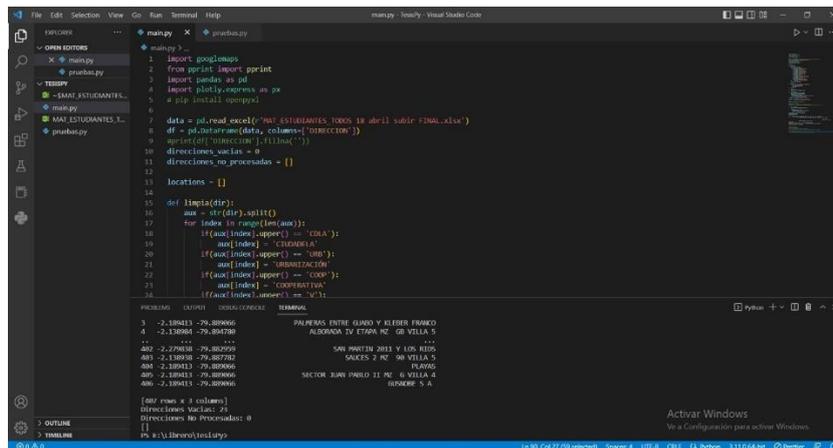


Figura 29 Utilización de la Librería PPrint

4.3.4 Pandas

Con esta librería se creó un tipo de dato conocido como dataframe, y este a su vez se compone de filas y columnas; este se utilizó para poner los datos de Excel en el gráfico. Pandas es una librería de Python especializada en el manejo y análisis de estructuras de datos. Pandas dispone de tres estructuras de datos diferentes:

- Series: Estructura de una dimensión.
- DataFrame: Estructura de dos dimensiones (tablas).
- Panel: Estructura de tres dimensiones (cubos).

Estas estructuras se construyen a partir de arrays de la librería NumPy, añadiendo nuevas funcionalidades.

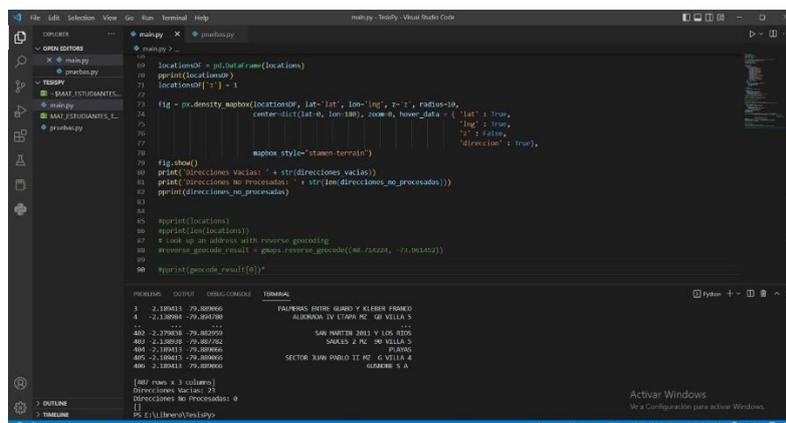


Figura 30 Pantalla con función DataFrame de Pandas

4.3.5 Dash

Sirve para hacer gráficos interactivos mediante el uso de eventos. A través de un par de patrones simples, Dash abstrae todas las tecnologías y protocolos que se requieren para crear una aplicación web completa con visualización de datos interactiva. Las aplicaciones de Dash se representan en el navegador web. Dado que las aplicaciones de Dash se ven en el navegador web, Dash es inherentemente multiplataforma y está listo para dispositivos móviles. Con todo lo mencionado Dash se convierte en la mejor opción para graficar el mapa de calor por su interactividad con otras librerías que se van a usar.

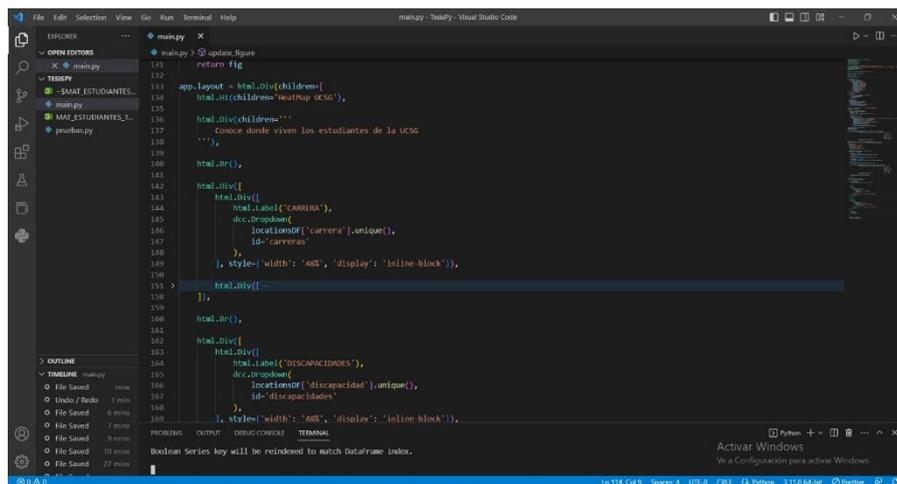


Figura 31 Dash 1

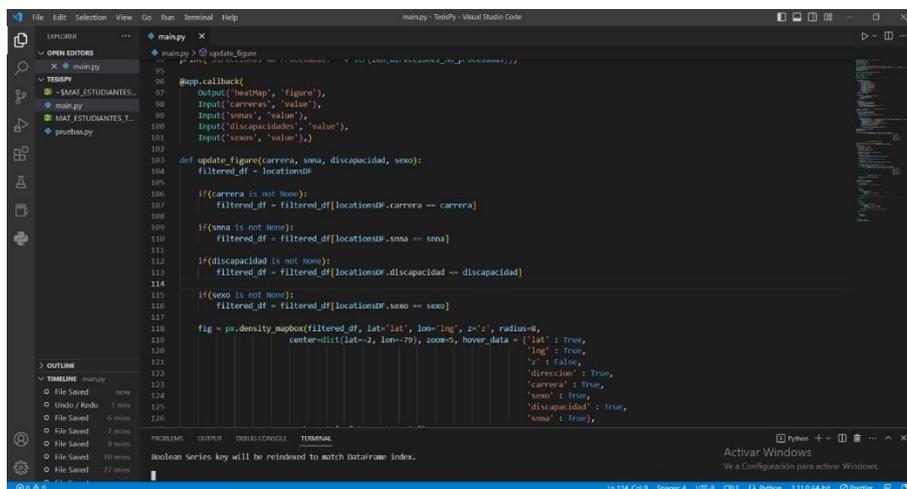


Figura 32 Dash 2

HeatMap UCSG

Conoce donde viven los estudiantes de la UCSG

CARRERA

Select...

DISCAPACIDADES

Select...

CANTIDAD DE DIRECCIONES PROCESADAS

407

CANTIDAD DE DIRECCIONES VACIAS

23

CANTIDAD DE DIRECCIONES NO PROCESADAS

0

BECA SNNA

NO SI

SEXO

HOMBRE MUJER



Figura 33 Dash 3

4.4 Tipos de Datos

4.4.1 Flotantes

El tipo de datos FLOAT almacena números de punto flotante de doble precisión con hasta 17 dígitos significativos. FLOAT corresponde al punto flotante de 4 bytes de IEEE y al tipo de datos doble en C. El rango de valores para el tipo de datos FLOAT es el mismo que el rango del tipo de datos doble C en su computadora. En la implementación del programa los datos que tenían este formato eran la latitud y la longitud.

4.4.2 String

Un tipo de datos de cadena es tradicionalmente una secuencia de caracteres, ya sea como una constante literal o como algún tipo de variable. Este último puede permitir la mutación de sus elementos y el cambio de longitud, o puede ser fijo (después de la creación). Una cadena generalmente se considera un tipo de datos y, a menudo, se implementa como una estructura de datos de matriz de bytes (o palabras) que almacena una secuencia de elementos,

generalmente caracteres, utilizando alguna codificación de caracteres. Los datos de este tipo en la herramienta son la dirección y la carrera de los estudiantes.

4.4.3 Dataframe

Se puede hacer una analogía con una tabla. El dataframe posee tanto filas como columnas; usualmente los dataframes se construyen a partir de diccionarios, es decir, la llave de diccionario va a ser la columna y el valor va a ocupar la posición de las filas.

CONCLUSIONES

Desde el preciso instante en que se inició con la ejecución de este trabajo hasta su término se ha llegado a la determinación de que el programa informático de mapa de calor cumple con los diversos objetivos planteados con anterioridad:

- Se obtuvo de manera satisfactoria los datos que fueron utilizados para la correcta ejecución y desarrollo del presente proyecto. Cabe recalcar que los datos conseguidos estaban en un estado muy básico, sin embargo, así era la forma en la que se esperaba tenerlos en un inicio.
- Debido al estado inicial de los datos, se procedió a hacer las respectivas etapas de limpieza y preparación de datos a través de diversas herramientas de filtración y correjimiento de información. De esta forma los datos estuvieron listos para ser utilizados en su propósito original.
- Se realizó la correcta implementación del programa informático que permitió mostrar las ubicaciones de procedencia y de residencia de los alumnos de la facultad de Ingeniería de la UCSG.
- Con la herramienta ya preparada, se procedió con la construcción de un mapa de calor que muestra la demografía perteneciente a los alumnos de la Universidad Católica de Santiago de Guayaquil.

RECOMENDACIONES

- Una mejora del proyecto se encuentra en una de sus principales características, y esta tiene que ver con los datos. En el conjunto de prueba utilizado, si bien el principal dato, es decir, la dirección del estudiante se encuentra en el conjunto, una posible mejora sería observar más datos de aquel estudiante tal como su ciclo, teléfono e incluso edad.
- El tratamiento de datos forma la parte más importante del proyecto, ya que sin tratar los datos no se podrían mostrar en el aplicativo. En el conjunto de prueba se usaron diferentes algoritmos de limpieza de datos, sin embargo, las direcciones de los estudiantes son ingresadas manualmente, lo cual significa que el usuario siempre podrá ingresar caracteres no deseados en la dirección del estudiante. Esto se traduce en mejorar los algoritmos de limpieza constantemente para que el programa siga leyendo y procesando todas las direcciones.
- Las direcciones deben ser ingresadas haciendo un contraste con la encontrada en Google Maps, el estudiante debe señalar el punto exacto en donde reside usando la aplicación y la dirección que retorna ese punto es la que se ingresa en el sistema.

REFERENCIAS

- Anunaya, S. (2021, agosto 10). Data Preprocessing in Data Mining -A Hands On Guide. *Analytics Vidhya*.
<https://www.analyticsvidhya.com/blog/2021/08/data-preprocessing-in-data-mining-a-hands-on-guide/>
- Aparicio, M., & Costa, C. J. (2015). Data visualization. *Communication Design Quarterly*, 3(1), 7–11. <https://doi.org/10.1145/2721882.2721883>
- Arnold, K., Gosling, J., & Holmes, D. (2005). *THE Java™ Programming Language, Fourth Edition*. 642.
- Bal, H. E., Steiner, J. G., & Tanenbaum, A. S. (1989). Programming languages for distributed computing systems. *ACM Computing Surveys*, 21(3), 261–322. <https://doi.org/10.1145/72551.72552>
- Bhaskar, S., Kop, C., & Simonsen, J. G. (2022). Subclasses of Ptime Interpreted by Programming Languages. *Theory of Computing Systems*. <https://doi.org/10.1007/s00224-022-10074-z>
- Boyd, W. (2022, junio 22). *What is the Javascript programming language?* A Cloud Guru. <https://acloudguru.com/blog/engineering/what-is-the-javascript-programming-language>
- Bracha, G. (2004). *Generics in the Java Programming Language*. 23.
- Buttice, C. (2021, octubre 20). *What is C++ Programming Language? - Definition from Techopedia*. Techopedia.Com.
<http://www.techopedia.com/definition/26184/c-plus-plus-programming-language>
- Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., & Riddell, A. (2017). Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, 76(1). <https://doi.org/10.18637/jss.v076.i01>
- Chand, M. (2020, marzo 7). *What Is C#*. <https://www.c-sharpcorner.com/article/what-is-c-sharp/>
- Chen, C., Härdle, W. K., & Unwin, A. (2007). *Handbook of Data Visualization*. Springer Science & Business Media.

- Coursera. (2022, noviembre 14). *What Is Python Used For? A Beginner's Guide*. Coursera. <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>
- Courtney, A. (1995). *Phantom: An Interpreted Language for Distributed Programming*. USENIX Conference on Object-Oriented Technologies (COOTS 95). <https://www.usenix.org/conference/coots-95/phantom-interpreted-language-distributed-programming>
- Ernest, A., Mensah, E., & Gilbert, A. (2017). Qualitative Assessment of Compiled, Interpreted and Hybrid Programming Languages. *Communications on Applied Electronics*, 7(7), 8–13. <https://doi.org/10.5120/cae2017652685>
- Famili, A., Shen, W.-M., Weber, R., & Simoudis, E. (1997). Data Preprocessing and Intelligent Data Analysis. *Intelligent Data Analysis*, 1(1), 3–23. <https://doi.org/10.3233/IDA-1997-1102>
- Friendly, M. (2008). A Brief History of Data Visualization. En C. Chen, W. Härdle, & A. Unwin (Eds.), *Handbook of Data Visualization* (pp. 15–56). Springer. https://doi.org/10.1007/978-3-540-33037-0_2
- García, S., Luengo, J., & Herrera, F. (2015). *Data Preprocessing in Data Mining* (Vol. 72). Springer International Publishing. <https://doi.org/10.1007/978-3-319-10247-4>
- García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: Methods and prospects. *Big Data Analytics*, 1(1), 9. <https://doi.org/10.1186/s41044-016-0014-0>
- Gehlenborg, N., & Wong, B. (2012). Heat maps. *Nature Methods*, 9(3), 213–213. <https://doi.org/10.1038/nmeth.1902>
- Grimes, D. A., & Schulz, K. F. (2002). Descriptive studies: What they can and cannot do. *The Lancet*, 359(9301), 145–149. [https://doi.org/10.1016/S0140-6736\(02\)07373-7](https://doi.org/10.1016/S0140-6736(02)07373-7)
- Hao, J., & Ho, T. K. (2019). Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language. *Journal of Educational and Behavioral Statistics*, 44(3), 348–361. <https://doi.org/10.3102/1076998619832248>
- Hartman, J. (2020, enero 6). *What is Java? Definition, Meaning & Features of Java Platforms*. <https://www.guru99.com/java-platform.html>

- Hejlsberg, A., Torgersen, M., Wiltamuth, S., & Golde, P. (2008). *The C# Programming Language*. Pearson Education.
- Hejlsberg, A., Wiltamuth, S., & Golde, P. (2003). *C# Language Specification*. Addison-Wesley Longman Publishing Co., Inc.
- Helmus, J. J., & Collis, S. M. (2016). The Python ARM Radar Toolkit (Py-ART), a library for working with weather radar data in the Python programming language. *Journal of Open Research Software*, 4. <https://doi.org/10.5334/jors.119>
- Hope, C. (2021, marzo 13). *What is a Programming Language?* <https://www.computerhope.com/jargon/p/programming-language.htm>
- Iverson, K. E. (1962). A programming language. *Proceedings of the May 1-3, 1962, spring joint computer conference*, 345–351. <https://doi.org/10.1145/1460833.1460872>
- Jamshed, H., Khan, S. A., Khurram, M., Inayatullah, S., & Athar, S. (2019). Data Preprocessing: A preliminary step for web data mining. *3c Tecnología: Glosas de Innovación Aplicadas a La Pyme*, 8(Extra 1), 206–221.
- Kamiran, F., & Calders, T. (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1), 1–33. <https://doi.org/10.1007/s10115-011-0463-8>
- Koh, E. T., & Owen, W. L. (2000). Descriptive Research and Qualitative Research. En E. T. Koh & W. L. Owen (Eds.), *Introduction to Nutrition and Health Research* (pp. 219–248). Springer US. https://doi.org/10.1007/978-1-4615-1401-5_12
- Lake, L. (2019, noviembre 30). *What Is a Target Audience?* The Balance. <https://www.thebalancemoney.com/what-is-a-target-audience-2295567>
- Masco, J. (2008). Target audience. *Bulletin of the Atomic Scientists*, 64(3), 22–45. <https://doi.org/10.1080/00963402.2008.11461154>
- Nassaji, H. (2015). Qualitative and descriptive research: Data type versus data analysis. *Language Teaching Research*, 19(2), 129–132. <https://doi.org/10.1177/1362168815572747>
- Oussous, A., Benjelloun, F.-Z., Ait Lahcen, A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences*, 30(4), 431–448. <https://doi.org/10.1016/j.jksuci.2017.06.001>

- Pedamkar, P. (2019, marzo 28). What is Big Data Technology? | Complete Guide to Big Data Technology. *EDUCBA*. <https://www.educba.com/what-is-big-data-technology/>
- Post, F. H., Nielson, G., & Bonneau, G.-P. (2002). *Data Visualization: The State of the Art*. Springer Science & Business Media.
- Rassokhin, D. (2020). The C++ programming language in cheminformatics and computational chemistry. *Journal of Cheminformatics*, 12(1), 10. <https://doi.org/10.1186/s13321-020-0415-y>
- Rubkiewicz, A. (2022, junio 9). *How to define heat maps and what types you can use for your business*. <https://livesession.io/heatmaps/what-is-a-heat-map/>
- Sanner, M. F. (s/f). *PYTHON: A PROGRAMMING LANGUAGE FOR SOFTWARE INTEGRATION AND DEVELOPMENT*. 8.
- Scarton, C., & Specia, L. (2018). Learning Simplifications for Specific Target Audiences. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 712–718. <https://doi.org/10.18653/v1/P18-2113>
- Schafmeister, C. E. (2016). CANDO: A Compiled Programming Language for Computer-Aided Nanomaterial Design and Optimization Based on Clasp Common Lisp. *Proceedings of the 9th European Lisp Symposium on European Lisp Symposium*, 75–82.
- Stevens, E. (2021, julio 15). *What Is Data Visualization? A Complete Introductory Guide*. <https://careerfoundry.com/en/blog/data-analytics/what-is-data-visualization/>
- Stroustrup, B. (1999). *An Overview of the C++ Programming Language*. 23.
- Svensson, L. (1984). *Three Approaches to Descriptive Research*.
- Theisen, K. J. (2019). Programming languages in chemistry: A review of HTML5/JavaScript. *Journal of Cheminformatics*, 11(1), 11. <https://doi.org/10.1186/s13321-019-0331-1>
- TopUniversities, E. de. (s/f). *PUniversidad Católica de Santiago de Guayaquil*. Top Universities. Recuperado el 1 de diciembre de 2022, de <https://www.topuniversities.com/universities/puniversidad-catolica-de-santiago-de-guayaquil>

- Universidades de Ecuador, E. de. (s/f). *Universidades de Ecuador*.
Universidades de Ecuador. Recuperado el 1 de diciembre de 2022, de
<http://universidades.com.ec/>
- Waskom, M. (2021). seaborn: Statistical data visualization. *Journal of Open Source Software*, 6(60), 3021. <https://doi.org/10.21105/joss.03021>
- Wilkinson, L., & Friendly, M. (2009). The History of the Cluster Heat Map. *The American Statistician*, 63(2), 179–184.
<https://doi.org/10.1198/tas.2009.0033>
- Wirfs-Brock, A., & Eich, B. (2020). JavaScript: The first 20 years. *Proceedings of the ACM on Programming Languages*, 4(HOPL), 77:1-77:189.
<https://doi.org/10.1145/3386327>
- Zuiderwijk, A., Janssen, M., & Dwivedi, Y. K. (2015). Acceptance and use predictors of open data technologies: Drawing upon the unified theory of acceptance and use of technology. *Government Information Quarterly*, 32(4), 429–440. <https://doi.org/10.1016/j.giq.2015.09.005>

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Héctor Xavier Ramírez Vega**, con C.C: # **0931924260** autor/a del trabajo de titulación: **“Prototipo de mapa de calor para identificar la demografía descriptiva de distribución geográfica de los estudiantes de la facultad de Ingeniería de la Universidad Católica de Santiago de Guayaquil.”** previo a la obtención del título de **Ingeniero en Ciencias de la Computación** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 16 de febrero de 2023



Nombre: **Héctor Xavier Ramírez Vega**

C.C: **0931924260**



REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TEMA Y SUBTEMA:	Prototipo de mapa de calor para identificar la demografía descriptiva de distribución geográfica de los estudiantes de la facultad de Ingeniería de la Universidad Católica de Santiago de Guayaquil.		
AUTOR(ES)	Héctor Xavier Ramírez Vega		
REVISOR(ES)/TUTOR(ES)	César Adriano Salazar Tovar		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Ingeniería		
CARRERA:	Ingeniería en Ciencias de la Computación		
TITULO OBTENIDO:	Ingeniero en Ciencias de la Computación		
FECHA DE PUBLICACIÓN:	16 de febrero de 2023	No. DE PÁGINAS:	67
ÁREAS TEMÁTICAS:	Procesamiento de Datos, Visualización de Datos, Mapa de Calor		
PALABRAS CLAVES/ KEYWORDS:	Lenguaje de programación, procesamiento de datos, visualización de datos, demografía, geografía, mapa.		
RESUMEN/ABSTRACT:	<p>El actual proyecto de integración curricular posee un objetivo primordial, el cual es proponer una herramienta de ayuda para la Universidad Católica de Santiago de Guayaquil, en la cual se pueda presentar información relacionada con el lugar de procedencia y residencia de los estudiantes de esta institución. A lo largo de la ejecución del presente trabajo se tuvo muy en cuenta conceptos relacionados con los lenguajes de programación, procesamiento y visualización de datos; así como también tener presente varios aspectos relacionados con la UCSG, que es la empresa sobre la cual se está realizando este proyecto. El método de investigación empleado fue el descriptivo, ya que para hacer un progreso fue muy necesario obtener datos de una persona cuyo trabajo incluya el hecho de hacer revisiones constantes de información de los estudiantes. Esta recopilación de datos se la realizó a través de la técnica de recolección de información conocida como entrevista. La finalidad de esto era conocer que es lo que se podía hacer con la información disponible, así como con aquella extra. El desarrollo de la propuesta tecnológica dependió en gran manera de la actualidad con respecto a los datos relacionados con el lugar de procedencia y residencia de los estudiantes de la Facultad de Ingeniería de la UCSG, así como de lo recolectado en la entrevista. Todo esto con el objetivo de poder desarrollar una herramienta tecnológica que cumpla con las necesidades presentes en la actualidad con respecto a datos geográficos de los alumnos. Con la creación de este software informático se tiene como expectativa poder ayudar a que quienes toman decisiones de peso en la UCSG, puedan hacerlo de una mejor manera teniendo de primera mano esta información de procedencia y residencia de los estudiantes.</p>		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTORES:	Teléfono: +593-994-504405	E-mail: hectorxavierrv@outlook.com	
CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE)::	Toala Quimí, Edison José		
	Teléfono: +593-990-976776		
	E-mail: edison.toala@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			