

Desarrollo de Aplicación Web para Almacenamiento y Análisis de Datos de la Industria  
Petrolera Mundial Utilizando Herramientas TIC

Guillermo José Orozco Rugeles

Trabajo de Grado para Optar al Título de Ingeniero de Petróleos

Director

Germán González Silva, Ph.D

Universidad Industrial de Santander  
Facultad de Ingenierías Físicoquímicas  
Escuela de Ingeniería de Petróleos  
Programa Académico  
Bucaramanga

2024

### **Dedicatoria**

A Dios, mi guía y fuente inagotable de fortaleza y sabiduría. Sin Su luz y bendición, este logro no habría sido posible. Le agradezco por cada día, por cada oportunidad y por cada reto que me ha permitido superar.

A mi amada Aura María Manjarres Hernández, cuyo amor y apoyo incondicional han sido mi refugio y mi motivación constante. Gracias por estar a mi lado en cada paso de este camino, por creer en mí y por compartir este sueño conmigo. Tu presencia ilumina mi vida y me inspira a ser mejor cada día.

A mi familia, el pilar fundamental de mi vida. A mi madre, Martha Rugeles, por su amor incondicional, su sabiduría y su ejemplo de perseverancia. A mi padre, Luis Orozco, por enseñarme el valor del esfuerzo y la dedicación. A mis hermanas, Daniela Orozco y María Andrea Orozco, por su cariño, su complicidad y por ser siempre un apoyo inquebrantable. Cada uno de ustedes ha contribuido de manera invaluable a mi crecimiento personal y profesional.

A mi segunda familia, Jairo Manjarres y Elda Rubiela Rojas, por acogerme con tanto amor y generosidad. Su apoyo y afecto han sido un regalo invaluable en mi vida, y me siento afortunado de contar con ustedes.

Este logro es tanto mío como de todos ustedes. Gracias por su amor, por su fe en mí y por ser parte de este viaje. Su apoyo ha sido mi mayor fortaleza y motivación.

### **Agradecimientos**

Quiero expresar mi más sincero agradecimiento a todas las personas que han contribuido de manera significativa al desarrollo de este trabajo de grado.

En primer lugar, agradezco profundamente a mi director de tesis, el Dr. Germán González Silva, por su invaluable guía, apoyo constante y experticia a lo largo de todo este proceso. Su visión y conocimientos han sido fundamentales para dar forma y dirección a este proyecto.

Extiendo mi gratitud al ingeniero Carlos Eduardo Rugeles por su invaluable ayuda en la estructuración del código y su guía constante durante el desarrollo de la aplicación. Su experiencia y conocimientos fueron fundamentales para superar los desafíos técnicos que surgieron durante el proceso.

Agradezco especialmente a Aura María Manjarres Hernández, cuyo apoyo incondicional y paciencia fueron pilares fundamentales durante todo este proceso. Su ayuda en la lectura crítica del documento y sus valiosas sugerencias contribuyeron significativamente a mejorar la calidad del trabajo final.

Mi sincero agradecimiento a mi madre, Martha Rugeles, y a mi padre, Luis Orozco, por su apoyo inquebrantable y su asistencia en la revisión y corrección del documento. Sus perspectivas y comentarios fueron esenciales para refinar el contenido y la presentación de esta tesis.

Un agradecimiento especial a Ruby Rojas, cuya dedicación en la lectura del documento y sus observaciones detalladas ayudaron a pulir y mejorar significativamente el trabajo final.

También quiero reconocer a la Universidad Industrial de Santander y a la Facultad de Ingenierías Físicoquímicas por proporcionar los recursos y el ambiente académico necesarios para llevar a cabo esta investigación.

Finalmente, agradezco a todos los profesores, compañeros y amigos que, de una u otra forma, contribuyeron con sus conocimientos, apoyo y ánimo durante el desarrollo de este trabajo de grado.

A todos ellos, mi más sincero agradecimiento por hacer posible este logro académico y profesional.

**Tabla de Contenido**

	<b>Pág.</b>
Introducción .....	19
1. Objetivos .....	21
1.1 Objetivo General .....	21
1.2 Objetivos Específicos.....	21
2. Justificación .....	21
3. Estructura del Trabajo de Grado .....	21
3.1 Capitulo 1 – Conceptos Generales y Estado del Arte .....	23
3.1.1 Big Data Analytics .....	23
3.1.2 Bases de datos .....	24
3.1.3 Sistema de gestión de bases de datos (SGBD).....	24
MySQL .....	26
PostgreSQL.....	26
Microsoft SQL Server.....	27
Oracle Database .....	28
MongoDB .....	28
SQLite .....	29
MariaDB .....	30
Cassandra .....	30
3.1.4 Lenguajes de programación .....	31
Python .....	33
Java .....	34

C#.....	35
C++ .....	36
JavaScript.....	37
3.1.5 Softwares de programación.....	38
Adobe Dreamweaver .....	40
Lazarus .....	41
Delphi.....	42
Eclipse.....	43
IBM COBOL .....	43
PyCharm .....	44
Netbeans IDE .....	45
GeneXus.....	46
CodeLite.....	47
Atom .....	47
Flutter .....	48
Ionic .....	49
React Native.....	50
Next.js .....	50
3.1.6 Visualización de datos .....	51
Gráfico de barras.....	52
Gráfico de líneas .....	53
Gráfico de pastel .....	54
Histograma .....	55

Gráfico de Dispersión .....	56
Gráfico de Área.....	57
Gráfico de Radar .....	58
Gráfico de Burbuja.....	59
3.2 Capitulo 2 – Selección de Herramientas y Base de Datos .....	60
3.2.1 Lenguaje de Programación .....	60
3.2.2 Normalizador y Persistencia de Datos .....	62
3.2.3 Base de Datos.....	63
3.2.4 Construcción de la interfaz .....	64
3.2.5 Entorno de Desarrollo Integrado.....	65
3.2.6 Navegador de Preferencia .....	67
3.3 Capitulo 3 – Desarrollo de la Aplicación.....	68
3.3.1 Desarrollo del Cliente .....	70
3.3.1.1 Components .....	70
Chart.tsx .....	71
SelectorBar.tsx .....	71
Sidebar.tsx y SidebarMenuItem.tsx .....	71
visor.tsx.....	71
Wrapper.tsx.....	71
3.3.1.2 Carpetas Fuentes de Energía.....	72
3.3.2 Configuración del Servidor.....	73
3.3.3 Desarrollo del Controlador .....	75
3.3.4 Pruebas Beta.....	76

3.3.4.1 Comparación de graficas.....	78
Primary Energy: Consumption – Exajoules (from 1965) – North America .....	79
Carbon Dioxide Emissions from Flaring – Million tonnes of carbon dioxide (from 1975) – North America.....	80
Oil: Production - Barrels (from 1965) – North America .....	81
Gas: Production – Bcf (from 1970) – North America .....	82
Coal: Production – Exajoules (from 1981) – North America.....	83
Nuclear Energy - Generation – TWh (from 1965) – North America .....	84
Hydroelectricity - Generation – TWh (from 1965) – North America .....	85
Renewables – Renewable Power – Exajoules (from 1965) – North America.....	86
Electricity generation - TWh (from 1985) – North America.....	87
Key materials - Lithium Production Thousand tonnes (from 1995) – South & Central America	88
3.3.4.2 Resultados de la prueba beta.....	89
3.4 Capitulo 4 – Resultados, Beneficios y Manual de Usuario .....	90
3.4.1 Resultados .....	90
3.4.2 Beneficios .....	91
3.4.3 Manual de Usuario.....	93
1. Como Ingresar al Programa .....	94
2. Partes del Programa .....	95
2.1 Side Nav.....	95
2.1.1 Primary Energy .....	95
2.1.2 Carbon Dioxide.....	96
2.1.3 Oil .....	96

2.1.4 Gas .....	97
2.1.5 Coal.....	97
2.1.6 Nuclear Energy .....	97
2.1.7 Hydroelectricity .....	98
2.1.8 Renewables .....	98
2.1.9 Electricity Generation .....	98
2.1.10 Key Materials.....	98
2.2 Filtros .....	99
2.2.1 Stat Filter.....	99
2.2.2 Region Filter .....	101
2.2.3 Graph Region/Country Filter .....	103
2.2.3.1 Graph Region Filter .....	103
2.2.3.2 Graph Country Filter.....	105
2.3 Display Zone.....	106
3 Como Graficar .....	107
3.5 Capitulo 5 – Conclusiones .....	109
3.6 Capitulo 6 – Recomendaciones y Futuras Mejoras.....	111
3.7 Capitulo 7 – Unidades y Factores de Conversión Aproximados .....	114
3.7.1 Unidades .....	114
Exajoules [EJ].....	114
Gigajoule per capita [GJ/cap] .....	115
Million Tonnes of carbon dioxide [MteCO <sub>2</sub> ].....	115
Billion cubic meters [bcm].....	115

Million Tonnes of carbon dioxide equivalent [MteCO <sub>2e</sub> ] .....	115
Thousand barrels per day [Kb/d] .....	116
Million tonnes [Mte] .....	116
Billion cubic feet per day [bcf/d] .....	116
Terawatt hour [Twh] .....	116
Thousand barrels of oil equivalent per day [Kboe/d] .....	117
Petajoules [PJ].....	117
Thousand tonnes [kt].....	117
Million Tonnes of Oil Equivalent [Mtoe].....	117
Trillion British Thermal Units [TBtu] .....	118
Million Barrels of Oil Equivalent [Mboe] .....	118
3.7.2 Factores de Conversión Aproximados .....	118
Crude Oil.....	118
Oil Products .....	119
Natural gas (NG) and liquefied natural gas (LNG) .....	119
Thermal equivalent efficiency factors used to convert non-fossil electricity to primary energy	120
Factores de Conversión Aproximados para Unidades .....	121
Equivalentes Caloríficos .....	121
Valores Caloríficos Netos o Inferiores .....	122
Referencias Bibliográficas .....	123

**Lista de Tablas**

	<b>Pág.</b>
Tabla 1 – <i>Factores de Conversión del Petróleo</i> .....	118
Tabla 2 – <i>Factores de Conversión de Subproductos del Petróleo</i> .....	119
Tabla 3 – <i>Factores de Conversión del Gas Natural y el Gas Natural Licuad</i> .....	119
Tabla 4 – <i>Factores de Eficiencia</i> .....	120
Tabla 5 – <i>Equivalencias Caloríficas de un Exajulio</i> .....	121

**Lista de Figuras**

	<b>Pág.</b>
Figura 1 – <i>Ejemplo de gráfico de barras</i> .....	53
Figura 2 – <i>Ejemplo de gráfico de líneas</i> .....	54
Figura 3 – <i>Ejemplo de gráfico de pastel</i> .....	55
Figura 4 – <i>Ejemplo de gráfico histograma</i> .....	56
Figura 5 – <i>Ejemplo de grafico de dispersión</i> .....	57
Figura 6 – <i>Ejemplo de gráfico de área</i> .....	58
Figura 7 – <i>Ejemplo de gráfico de radar o araña</i> .....	59
Figura 8 – <i>Ejemplo de gráfico de burbuja</i> .....	60
Figura 9 – <i>Comparativo consumo energías primarias en Exajulios desde 1965 para Norte América</i> .....	79
Figura 10 – <i>Comparativo emisiones de dióxido de carbono provenientes de la quema en antorcha en millones de toneladas desde 1975 para Norte América</i> .....	80
Figura 11 – <i>Comparativo producción de petróleo en miles de barriles desde 1965 para Norte América</i> .....	81
Figura 12 – <i>Comparativo producción de gas en miles de millones de pies cúbicos desde 1970 para Norte América</i> .....	82
Figura 13 – <i>Comparativo producción de carbón en Exajulios desde 1981 para Norte América</i>	83
Figura 14 – <i>Comparativo generación de energía nuclear en Teravatio-hora desde 1965 para Norte América</i> .....	84

Figura 15 – *Comparativo generación de hidroelectricidad en Teravatio-hora desde 1965 para Norte América.* ..... 85

Figura 16 – *Comparativo generación de energía renovable en Exajulios desde 1965 para Norte America.* ..... 86

Figura 17 – *Comparativo generación de electricidad en Teravatio-hora desde 1985 para Norte América* ..... 87

Figura 18 – *Comparativo producción de Litio en miles de toneladas desde 1995 para Norte América.* ..... 88

## Glosario

**API:** interfaz de Programación de Aplicaciones, conjunto de protocolos y herramientas para construir aplicaciones de software.

**API REST:** interfaz de Programación de Aplicaciones que utiliza HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos en formatos como JSON o XML.

**Big Data Analytics:** proceso de examinar grandes volúmenes de datos para descubrir patrones ocultos, correlaciones y otras informaciones útiles.

**BP Statistical Review:** base de datos confiable de estadísticas energéticas globales utilizada en la aplicación.

**Componente:** parte modular y reutilizable de una interfaz de usuario en React.

**Controlador:** componente que maneja la lógica de negocio y la comunicación entre el cliente y la base de datos.

**CSV:** formato de archivo de valores separados por comas, utilizado para almacenar datos tabulares.

**Depuración:** proceso de identificar y corregir errores en el código de un programa.

**Escalabilidad:** capacidad de un sistema para manejar una carga creciente de trabajo o su potencial para ser ampliado.

**Exajulio (EJ):** unidad de medida de energía equivalente a  $10^{18}$  julios.

**Framework:** marco de trabajo que proporciona una estructura y conjunto de herramientas para el desarrollo de software.

**Gigajulio per cápita (GJ/cap):** unidad de medida que expresa el consumo de energía por persona.

**Git:** sistema de control de versiones distribuido para rastrear cambios en el código fuente durante el desarrollo de software.

**HTTP:** protocolo de transferencia de hipertexto, base de la comunicación de datos en la World Wide Web.

**IDE (Entorno de Desarrollo Integrado):** software que proporciona herramientas para facilitar el desarrollo de aplicaciones.

**JavaScript:** lenguaje de programación utilizado principalmente para crear páginas web interactivas.

**JSON:** formato ligero de intercambio de datos, fácil de leer y escribir para humanos y máquinas.

**Middleware:** software que actúa como puente entre un sistema operativo o base de datos y las aplicaciones, especialmente en una red.

**MongoDB:** sistema de gestión de bases de datos NoSQL utilizado en la aplicación para almacenar y manejar datos.

**Next.js:** framework de React utilizado para el desarrollo de la interfaz de usuario de la aplicación.

**NoSQL:** enfoque de diseño de bases de datos que permite el almacenamiento y la recuperación de datos modelados de formas diferentes a las tablas relacionales tradicionales.

**NPM (Node Package Manager):** gestor de paquetes para JavaScript, utilizado principalmente en Node.js.

**Optimización:** proceso de mejorar la eficiencia, capacidad y rendimiento de un sistema o aplicación.

**React:** biblioteca de JavaScript para construir interfaces de usuario.

**Refactorización:** proceso de reestructurar el código existente sin cambiar su comportamiento externo.

**Renderizado del lado del servidor (SSR):** técnica de renderizado de páginas web en el servidor antes de enviarlas al cliente.

**SSG (Static Site Generation):** técnica de generación de sitios web estáticos durante el tiempo de compilación.

**TIC (Tecnologías de la Información y la Comunicación):** conjunto de tecnologías desarrolladas para gestionar información y enviarla de un lugar a otro.

**TypeScript:** superconjunto tipado de JavaScript que compila a JavaScript puro.

**Visualización de datos:** representación gráfica de datos e información.

### Resumen

**Título:** Desarrollo de Aplicación Web para Almacenamiento y Análisis de Datos de la Industria Petrolera Mundial Utilizando Herramientas TIC\*

**Autor:** Guillermo José Orozco Rugeles\*\*

**Palabras Clave:** Análisis de Big Data, Gestión de Datos, Visualización de Datos, React, Almacenamiento de datos, Aplicación web, Base de datos, Integración de sistemas.

**Descripción:** Esta tesis se enfoca en el desarrollo de una aplicación web para el almacenamiento y análisis de datos de la industria petrolera mundial utilizando herramientas de Tecnologías de la Información y Comunicación (TIC). El proyecto aborda la creciente necesidad de gestionar y analizar grandes volúmenes de datos (Big Data) en el sector petrolero de manera eficiente y accesible.

La aplicación propuesta integrará tecnologías modernas como React para crear una interfaz de usuario dinámica y responsivo. Se implementarán técnicas avanzadas de gestión de datos y almacenamiento para manejar la complejidad y el volumen de información del sector. La visualización de datos jugará un papel crucial, permitiendo a los usuarios interpretar tendencias y patrones complejos de manera intuitiva.

El desarrollo se guiará por objetivos específicos que incluyen la identificación de variables clave en la industria petrolera, la selección de herramientas TIC, bases de datos apropiadas, y la aplicación de metodologías ágiles de desarrollo. Se pondrá especial énfasis en la integración de sistemas para asegurar una comunicación fluida entre los diferentes componentes de la aplicación.

El resultado será una herramienta robusta capaz de almacenar, procesar y visualizar datos críticos de la industria petrolera global. Esta aplicación no solo facilitará el análisis de Big Data en el sector, sino que también proporcionará perspectivas valiosas para la toma de decisiones estratégicas. Además, se desarrollará un manual de usuario detallado para garantizar la adopción efectiva de la herramienta en entornos profesionales.

---

\* Trabajo de Grado

\*\* Facultad de Ingenierías Fisicoquímicas. Escuela de Ingeniería de Petróleos. Ingeniería de Petróleos. Director: Germán González Silva, PhD.

### Abstract

**Title:** Development of Web Application for Storage and Analysis of Data from the Global Oil Industry Using ICT Tools \*

**Author(s):** Guillermo José Orozco Rugeles\*\*

**Key Words:** Big Data Analytics, Data Management, Data Visualization, React, Data storage, Web application, Database, Systems integration.

**Description:** This thesis focuses on developing a web application for storing and analyzing global oil industry data using Information and Communication Technology (ICT) tools. The project addresses the growing need to efficiently manage and analyze large volumes of data (Big Data) in the oil sector in an accessible manner.

The proposed application will integrate modern technologies such as React to create a dynamic and responsive user interface. Advanced data management and storage techniques will be implemented to manage the sector's complexity and volume of information. Data visualization will be crucial, allowing users to interpret complex trends and patterns intuitively.

The development will be guided by specific objectives including identifying key variables in the oil industry, selecting appropriate ICT tools and databases, and applying agile development methodologies. Special emphasis will be placed on system integration to ensure smooth communication between the different components of the application.

The result will be a robust tool capable of storing, processing, and visualizing critical data from the global oil industry. This application will facilitate Big Data analysis in the sector and provide valuable insights for strategic decision-making. Additionally, a detailed user manual will be developed to ensure the effective adoption of the tool in professional environments.

---

\* Degree Work

\*\*Faculty of Physical-Chemical Engineering. School of Petroleum Engineering. Petroleum Engineering. Director: Germán González Silva, PhD.

## Introducción

En la era digital actual, la industria petrolera se enfrenta a desafíos significativos en cuanto al manejo y análisis de grandes volúmenes de datos. La complejidad y diversidad de la información generada en las operaciones petroleras requieren herramientas sofisticadas para su almacenamiento, procesamiento y visualización efectiva (Mohammadpoor & Torabi, 2020). Las bases de datos son herramientas fundamentales para almacenar información crucial, desde datos de personal y pedidos hasta productos y consumo, proporcionando una visión general del desarrollo de los factores más importantes en una sociedad (Ricardo, 2009). Sin embargo, estas bases de datos suelen ser extensas y complejas, careciendo del dinamismo necesario para una lectura y análisis eficientes.

En la industria petrolera, una de las bases de datos más completas y relevantes es la "BP Statistical Review of World Energy", que contiene información desde 1965 hasta la actualidad en categorías como emisiones de CO<sub>2</sub>, petróleo, gas, carbón, energía nuclear, hidroelectricidad y energías renovables, abarcando datos de todos los países involucrados en cada sector (BP, 2022). Aunque estas bases de datos son vitales para comprender las tendencias globales, su verdadero valor radica en el análisis correspondiente. Sin un análisis adecuado, estos datos son meramente números sin contexto ni significado práctico (Sagiroglu & Sinanc, 2013).

Este proyecto de investigación aborda esta problemática mediante el desarrollo de una aplicación web innovadora para el almacenamiento y análisis de datos de la industria petrolera mundial utilizando herramientas de Tecnologías de la Información y Comunicación (TIC). El propósito principal es crear una herramienta que no solo almacene estos datos cruciales, sino que

también proporcione valores, gráficos y comparaciones de los diferentes parámetros relevantes de manera sencilla e intuitiva.

El sector petrolero genera diariamente enormes cantidades de datos provenientes de diversas fuentes. Sin embargo, muchas empresas aún carecen de sistemas integrados que permitan centralizar, procesar y analizar esta información de manera eficiente (Perrons & Jensen, 2015). Esto resulta en una subutilización de datos valiosos que podrían optimizar las operaciones y la toma de decisiones estratégicas. La aplicación propuesta integrará tecnologías modernas como React para crear una interfaz de usuario dinámica y adaptable, implementando técnicas avanzadas de gestión de datos y almacenamiento para manejar la complejidad y el volumen de información del sector (Facebook Inc., 2023).

Este estudio se fundamenta en la necesidad de proporcionar a las instituciones académicas, empresas petroleras y analistas del sector una herramienta robusta y accesible para manejar la complejidad de los datos de la industria. La justificación radica en su potencial para generar un impacto significativo tanto en el ámbito técnico como en el económico y social del sector petrolero, facilitando el acceso y análisis de información crítica, mejorando la eficiencia operativa y contribuyendo a una toma de decisiones más informada y transparente (Demirkan & Delen, 2013).

## **1. Objetivos**

### **1.1 Objetivo General**

Desarrollar una aplicación web para almacenamiento y análisis de datos de la industria petrolera mundial utilizando herramientas TIC.

### **1.2 Objetivos Específicos**

- I. Determinar las variables significativas o representativas del área a estudiar.
- II. Seleccionar las herramientas TIC y base de datos a utilizar.
- III. Aplicar metodologías para el desarrollo de la herramienta TIC.
- IV. Desarrollar el manual de usuario para el manejo de la herramienta TIC.

## **2. Justificación**

La necesidad de una plataforma de análisis de datos completa, sencilla, intuitiva y dinámica, generó la motivación para realizar este trabajo de grado, para así poder brindar a la comunidad académica y laboral una aplicación web que esté conectada a la base de datos “BP Statistical Review of World Energy” y que pueda otorgar de manera rápida y efectiva los valores solicitados por el usuario, además de poder compararlos con diferentes variables del entorno para así lograr un análisis y correlación entre ellos.

## **3. Estructura del Trabajo de Grado**

La tesis se estructura en varios capítulos que abordan el desarrollo de una aplicación web para el almacenamiento y análisis de datos en la industria petrolera mundial. En el primer capítulo, el marco teórico establece el contexto necesario para comprender el proyecto, explorando conceptos claves como el Big Data Analytics, Bases de datos, Gestión de Datos, Lenguajes de

programación, Softwares de programación y la visualización de datos. Estos conceptos son fundamentales para entender cómo se puede optimizar el manejo de grandes volúmenes de información en el sector petrolero, facilitando la toma de decisiones estratégicas.

El segundo capítulo describe la metodología utilizada para el desarrollo de la aplicación, detallando la selección de herramientas tecnológicas como React y bases de datos especializadas. Se explica la aplicación de metodologías ágiles que permiten un desarrollo interactivo y flexible, asegurando que los diferentes componentes tecnológicos se integren de manera efectiva. Este enfoque metodológico es crucial para garantizar que la aplicación cumpla con los requisitos específicos del sector.

En el tercer capítulo, se detalla el proceso de desarrollo de la aplicación web. Se cubren aspectos técnicos del diseño y la implementación, desde la programación hasta la configuración del entorno de desarrollo. Además, se describen las pruebas beta realizadas para asegurar que la aplicación funcione correctamente y cumpla con los objetivos planteados. Este capítulo es esencial para entender cómo se construyó la aplicación y cómo se garantiza su funcionalidad.

El cuarto capítulo presenta los resultados obtenidos tras la implementación de la aplicación. Se evalúa su eficacia en el almacenamiento y análisis de datos, destacando los beneficios que aporta al sector petrolero y educativo, como la optimización de procesos y la mejora en la toma de decisiones. Estos resultados demuestran el impacto positivo de la aplicación en la industria. Adicional a esto en el cuarto capítulo se muestra el manual de usuario para el uso de la aplicación web.

En el quinto y sexto capítulo se exponen las conclusiones y recomendaciones, se resumen los hallazgos principales y se ofrecen sugerencias para futuras mejoras. Se discute el impacto potencial de la aplicación y se sugieren áreas para investigaciones futuras, proporcionando una visión clara de cómo esta herramienta puede seguir evolucionando para satisfacer las necesidades del sector petrolero.

Por último se presenta el séptimo capítulo como un recurso esencial para la comprensión y manejo de los datos energéticos utilizados en la aplicación. Este capítulo proporciona una guía detallada de las diversas unidades de medida empleadas en la industria energética, incluyendo definiciones precisas y contextos de uso para unidades como Exajoules, Million Tonnes of Carbon Dioxide, y Billion cubic metres, entre otras. Además, ofrece una serie de factores de conversión aproximados que son cruciales para la interpretación y comparación de datos entre diferentes fuentes de energía y sistemas de medición. Este capítulo no solo sirve como referencia técnica para los usuarios de la aplicación, sino que también facilita la comprensión de los datos presentados, asegurando una interpretación precisa y coherente de la información energética global.

### **3.1 Capítulo 1 – Conceptos Generales y Estado del Arte**

#### ***3.1.1 Big Data Analytics***

Big Data Analytics es el proceso de examinar grandes volúmenes de datos para descubrir patrones ocultos, correlaciones desconocidas y otra información útil que puede ayudar a las organizaciones a tomar decisiones informadas (Sagiroglu & Sinanc, 2013). Este campo se ha vuelto esencial en la era digital debido al crecimiento exponencial de datos generados por diversas fuentes, como redes sociales, dispositivos IoT y transacciones comerciales (Mohammadpoor & Torabi, 2020). Las técnicas de análisis de Big Data utilizan algoritmos avanzados y herramientas

de procesamiento para analizar datos estructurados y no estructurados, permitiendo a las empresas mejorar sus operaciones y obtener una ventaja competitiva (Cui, Kara, & Chan, 2020).

El análisis de Big Data se aplica en diversos sectores, incluyendo la salud, el comercio minorista, las finanzas y la industria energética, donde ayuda a optimizar procesos, predecir tendencias del mercado y mejorar la experiencia del cliente (Ikegwu et al., 2022). Las herramientas de Big Data Analytics, como Hadoop y Spark, permiten procesar y analizar datos a gran escala de manera eficiente, mientras que las técnicas de aprendizaje automático y minería de datos ayudan a extraer información valiosa de estos conjuntos de datos masivos (Iqbal et al., 2020).

### **3.1.2 Bases de datos**

Se le considera base de datos a toda herramienta para recopilar y organizar información sobre personas, productos, pedidos u otras cosas y se pueden encontrar en cualquier ámbito de la vida cotidiana como visitas en un portal de internet del consumidor, servicio al cliente, banca electrónica, sistemas de reservación de cualquier hotel o aerolínea, etc.

En la actualidad, las bases de datos se usan tan ampliamente que se pueden encontrar en organizaciones de todos los tamaños, desde grandes corporaciones y agencias gubernamentales, hasta pequeños negocios e incluso en hogares. Las actividades diarias con frecuencia lo ponen en contacto con las bases de datos, ya sea directa o indirectamente. (Ricardo, 2009)

### **3.1.3 Sistema de gestión de bases de datos (SGBD)**

Es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, además de proporcionar un acceso controlado a la misma. Generalmente, un SGBD proporciona servicios como la definición de la base de datos mediante un lenguaje de definición de datos, lo

cual es bastante efectivo ya que permite especificar la estructura, el tipo de los datos y las restricciones sobre los mismos, también brinda la posibilidad de insertar, actualizar, eliminar y consultar cualquier tipo de datos.

Además de esto el SGBD proporciona un sistema de seguridad, de modo que los usuarios no autorizados no puedan acceder a la base de datos o simplemente limitar la interacción del usuario con la misma, de modo que estos no puedan modificarla y tengan simplemente permisos de visualización, proporciona un sistema de integridad que mantiene la misma y la consistencia de los datos, brinda un sistema de control de recuperación que restablece la base de datos después de que se produzca un fallo de hardware o del software, un diccionario de datos o catálogo, accesible por el usuario, que contiene la descripción de los datos de la base de datos. (Marques, 2011)

Los sistemas de gestión de bases de datos (SGBD) son herramientas clave que permiten almacenar, organizar y acceder a grandes volúmenes de datos de manera estructurada y segura (Silberschatz et al., 2020). A continuación, se presentan y describen algunos de los programas más populares en este ámbito:

- MySQL
- PostgreSQL
- Microsoft SQL Server
- Oracle Database
- MongoDB
- SQLite
- MariaDB

- Cassandra

### **MySQL**

MySQL es un sistema de gestión de bases de datos relacional de código abierto que es ampliamente utilizado en aplicaciones web y servicios en la nube debido a su velocidad, fiabilidad y facilidad de uso (MySQL, n.d.). Desarrollado inicialmente por MySQL AB y ahora propiedad de Oracle Corporation, MySQL permite a los usuarios almacenar y gestionar datos en tablas que pueden relacionarse entre sí, lo que facilita la organización y recuperación eficiente de información (Oracle, 2023).

Este sistema es conocido por su capacidad para manejar grandes volúmenes de datos y realizar consultas complejas de manera rápida y precisa. MySQL es compatible con múltiples plataformas, incluyendo Windows, Linux y macOS, y se integra bien con una variedad de lenguajes de programación como PHP, Java, y Python, lo que lo hace ideal para el desarrollo de aplicaciones web dinámicas (MySQL, n.d.).

Además, MySQL ofrece características avanzadas como soporte para transacciones, control de concurrencia y seguridad robusta, lo que lo convierte en una opción popular para aplicaciones empresariales críticas. Su comunidad activa y su extensa documentación facilitan la resolución de problemas y la implementación de soluciones personalizadas (Oracle, 2023).

### **PostgreSQL**

PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto, conocido por su robustez, extensibilidad y cumplimiento de estándares (PostgreSQL Global Development Group, n.d.). A menudo elegido para aplicaciones que requieren integridad de datos

y transacciones complejas, PostgreSQL ofrece características avanzadas como soporte para tipos de datos personalizados, índices avanzados y transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) (Stonebraker & Kemnitz, 1991). Su capacidad para manejar grandes volúmenes de datos y realizar consultas complejas de manera eficiente lo hace ideal para aplicaciones empresariales críticas (Codd, 1970).

Además, PostgreSQL es altamente extensible, permitiendo a los desarrolladores definir nuevos tipos de datos, funciones y operadores (Momjian, 2001). Esto, junto con su compatibilidad con múltiples plataformas, lo convierte en una opción versátil para una amplia gama de aplicaciones. También es compatible con lenguajes de programación como Python, Java y C++, lo que facilita su integración en diversos entornos de desarrollo (PostgreSQL Global Development Group, n.d.).

### **Microsoft SQL Server**

Microsoft SQL Server es un sistema de gestión de bases de datos relacional desarrollado por Microsoft, ampliamente utilizado en entornos empresariales para manejar grandes volúmenes de datos de manera eficiente y segura (Microsoft, 2023a). Este software es conocido por su capacidad para integrarse con otros productos de Microsoft, como Azure y Power BI, lo que facilita la creación de soluciones completas de análisis y visualización de datos (Patel, 2021). SQL Server ofrece características avanzadas como soporte para transacciones ACID, alta disponibilidad, y replicación de datos, lo que garantiza la integridad y disponibilidad de la información crítica para el negocio (Dewson, 2019). Además, proporciona herramientas de análisis y minería de datos que permiten a las organizaciones extraer insights valiosos de sus datos (Microsoft, 2023b). Con su robusta arquitectura de seguridad, SQL Server protege los datos contra accesos no autorizados,

haciendo de él una opción confiable para aplicaciones que requieren un alto nivel de seguridad y rendimiento (Ben-Gan, 2020).

### **Oracle Database**

Oracle Database es un sistema de gestión de bases de datos relacional desarrollado por Oracle Corporation, conocido por su capacidad para manejar grandes volúmenes de datos y su robustez en aplicaciones empresariales críticas (Oracle, 2023). Este software es ampliamente utilizado en sectores que requieren alta disponibilidad, seguridad y rendimiento, como la banca, telecomunicaciones y servicios financieros (Smith & Williams, 2022). Oracle Database ofrece características avanzadas como soporte para transacciones ACID, replicación de datos y particionamiento, lo que garantiza la integridad y disponibilidad de la información (Garcia-Molina, Ullman, & Widom, 2008). Además, su arquitectura escalable permite a las organizaciones ajustar el rendimiento del sistema según sus necesidades específicas, desde pequeñas aplicaciones hasta grandes infraestructuras de datos (Elmasri & Navathe, 2016). La capacidad de Oracle para integrar inteligencia artificial y análisis de datos en tiempo real lo convierte en una herramienta poderosa para la toma de decisiones estratégicas, permitiendo a las empresas obtener insights valiosos de sus datos de manera eficiente y segura (Stonebraker, 2018).

### **MongoDB**

MongoDB es un sistema de gestión de bases de datos NoSQL de código abierto que utiliza un modelo de datos basado en documentos (MongoDB, Inc., 2023). A diferencia de las bases de datos relacionales tradicionales, MongoDB almacena datos en documentos JSON flexibles, lo que permite una estructura de datos más dinámica y adaptable (Chodorow, 2013). Esto lo hace ideal

para aplicaciones que requieren escalabilidad horizontal y necesitan manejar grandes volúmenes de datos no estructurados o semiestructurados (Banker, 2011).

Una de las características más destacadas de MongoDB es su capacidad para escalar de manera eficiente a través de la distribución de datos en múltiples servidores, lo que garantiza alta disponibilidad y tolerancia a fallos (Sullivan, 2015). Además, ofrece potentes capacidades de consulta y agregación, permitiendo a los desarrolladores realizar análisis complejos de datos de manera sencilla (Dayley, 2014).

MongoDB es ampliamente utilizado en aplicaciones modernas, como plataformas de comercio electrónico, redes sociales y sistemas de gestión de contenido, donde la flexibilidad y la capacidad de manejar datos variados son esenciales (Membrey et al., 2010). Su integración con lenguajes de programación populares y su comunidad activa de desarrolladores contribuyen a su popularidad y continuo desarrollo en el ámbito de las bases de datos (Hows et al., 2015).

### **SQLite**

SQLite es un sistema de gestión de bases de datos relacional, ligero y de código abierto, que se utiliza ampliamente en aplicaciones móviles, dispositivos embebidos y navegadores web (Kreibich, 2010). A diferencia de otros sistemas de bases de datos que requieren un servidor separado, SQLite se integra directamente en la aplicación, lo que simplifica su configuración y uso (Owens, 2006). Esta característica lo hace ideal para aplicaciones que requieren una base de datos compacta y eficiente sin la necesidad de un servidor dedicado.

SQLite almacena toda la base de datos en un solo archivo, lo que facilita su distribución y copia de seguridad (Allen & Owens, 2010). A pesar de su simplicidad, ofrece características

avanzadas como soporte para transacciones ACID, integridad referencial y consultas complejas, lo que lo hace adecuado para una amplia gama de aplicaciones (Kreibich, 2010). Su compatibilidad con múltiples plataformas y su uso en proyectos de alto perfil, como navegadores web y sistemas operativos móviles, destacan su fiabilidad y versatilidad (SQLite, n.d.).

### **MariaDB**

MariaDB es un sistema de gestión de bases de datos relacional de código abierto, derivado de MySQL, que ofrece mejoras en el rendimiento y características adicionales (MariaDB Foundation, 2023). Desarrollado por la comunidad de software libre, MariaDB es conocido por su alta compatibilidad con MySQL, lo que facilita la migración de aplicaciones existentes (Bartholomew, 2015). Este sistema es ampliamente utilizado en aplicaciones web y empresariales debido a su capacidad para manejar grandes volúmenes de datos de manera eficiente (Schwartz et al., 2012). MariaDB incluye características avanzadas como motores de almacenamiento mejorados, soporte para transacciones ACID y replicación de datos, lo que garantiza la integridad y disponibilidad de la información (Bell et al., 2014). Además, su arquitectura escalable permite a las organizaciones ajustar el rendimiento del sistema según sus necesidades específicas, desde pequeñas aplicaciones hasta grandes infraestructuras de datos (Razzoli, 2014). La comunidad activa de MariaDB y su compromiso con la innovación continua lo convierten en una opción confiable y robusta para la gestión de bases de datos en diversos sectores (MariaDB Corporation, 2022).

### **Cassandra**

Apache Cassandra es un sistema de gestión de bases de datos NoSQL distribuido, diseñado para manejar grandes cantidades de datos en múltiples servidores sin un punto único de falla

(Lakshman & Malik, 2010). Desarrollado inicialmente por Facebook, Cassandra es conocido por su arquitectura altamente escalable y su capacidad para ofrecer alta disponibilidad y tolerancia a fallos, lo que lo hace ideal para aplicaciones críticas en tiempo real que requieren un rendimiento constante y fiable (Hewitt, 2010).

Cassandra utiliza un modelo de datos basado en columnas y ofrece una replicación flexible, permitiendo a los usuarios definir cómo y dónde se replican los datos en el clúster (DataStax, 2023). Esto asegura que los datos estén disponibles incluso en caso de fallos de hardware o interrupciones de red. Además, su modelo de consistencia eventual permite a las aplicaciones ajustar el equilibrio entre consistencia y disponibilidad según sus necesidades específicas (Abadi, 2012).

Este sistema es ampliamente utilizado en sectores como telecomunicaciones, finanzas y comercio electrónico, donde la capacidad de manejar grandes volúmenes de datos distribuidos es crucial (Carpenter & Hewitt, 2016). La comunidad activa de Cassandra y su compatibilidad con herramientas de análisis de datos como Apache Spark y Hadoop lo convierten en una opción poderosa para las organizaciones que buscan implementar soluciones de Big Data (Apache Software Foundation, 2023).

#### ***3.1.4 Lenguajes de programación***

Es necesario entender a que se refieren por lenguaje de programación, según Francisco Gortazar, Raquel Martínez y Víctor Fresno existen dos maneras de describir los lenguajes de programación para públicos diferentes, la forma coloquial y la formal, desde un punto de vista coloquial es una notación para comunicarle a una computadora lo que deseamos que se haga, de

manera un poco más formal, podemos definirlo como un sistema notacional para describir computaciones en una forma legible tanto para la maquina como para el ser humano.

En los lenguajes de programación se puede distinguir 5 generaciones principalmente:

- Primera generación. A esta generación pertenece el lenguaje máquina. Este consiste en una secuencia binaria de ceros y unos y es el único lenguaje que entienden las computadoras modernas.
- Segunda generación. A esta generación se le conoce como lenguajes ensambladores. Estos poseen una serie de reglas mnemotécnicas que hacen más sencilla la cultura y escritura de programas. Estas reglas consisten en simplemente asociar nombres legibles a cada una de las instrucciones soportadas por la maquina (ADD, SUB, LOAD, STORE, etc.)
- Tercera generación. A esta generación pertenecen lenguajes como C, FORTRAN o Java. Estos lenguajes ya son considerados de alto nivel, ya que están bastante alejados del lenguaje binario de la maquina y son mucho más legibles por el hombre. Es necesario entender que para que estos lenguajes los entienda la máquina, es necesario de unos programas complejos llamados compiladores. Estos traducen el lenguaje en código binario para que la maquina los entienda.
- Cuarta generación. Los lenguajes de esta cuarta generación son lenguajes con propósito específico, como SQL, NNATURAL o ABAP. Estos lenguajes fueron diseñados para solucionar problemas muy concretos.
- Quinta generación. Estos lenguajes son los utilizados principalmente en el área de la inteligencia artificial y se trata de lenguajes que permiten especificar

restricciones que se le indican al sistema, que resuelve un determinado problema sujeto a estas restricciones.

(Matinez Unanue, Gortazar Bellas, & Fresno Fernandez, 2016)

En general es un lenguaje de computadora que los programadores utilizan para comunicarse y desarrollar programas de software, aplicaciones, páginas web, scripts u otros conjuntos de instrucciones para que sean ejecutadas por los ordenadores. A través de los años, han existido innumerables lenguajes de programación, cada uno con sus ventajas y desventajas. En la actualidad, vivimos en una era tecnológica en la que cada día se intenta proporcionar la mejor tecnología, por eso cada vez aparecen nuevos lenguajes de programación que buscan corregir o mejorar los lenguajes de programación existentes (Villalba, Uruquia Moraleda, & Rubio Gonzales, 2021).

En el presente año los 5 lenguajes de programación que más son utilizados son:

- Python
- Java
- C#
- C++
- JavaScript

### **Python**

Python es un lenguaje de programación de alto nivel, ampliamente reconocido por su simplicidad y legibilidad (Lutz, 2013). Diseñado para ser fácil de entender y escribir, Python utiliza una sintaxis clara y concisa que permite a los desarrolladores enfocarse en resolver problemas en lugar de preocuparse por la complejidad del código (Van Rossum & Drake, 2009). Este lenguaje

es versátil y se utiliza en una variedad de aplicaciones, desde desarrollo web y análisis de datos hasta inteligencia artificial y automatización de tareas (McKinney, 2017).

Una de las características más destacadas de Python es su extensa biblioteca estándar, que proporciona módulos y paquetes para realizar tareas comunes, como manipulación de archivos, comunicación en red y pruebas de software, sin necesidad de escribir código desde cero (Python Software Foundation, 2023). Además, Python cuenta con una comunidad activa que contribuye a una amplia gama de bibliotecas y herramientas de terceros, lo que amplía aún más sus capacidades (Oliphant, 2007).

Python es un lenguaje interpretado, lo que significa que el código se ejecuta línea por línea, facilitando la depuración y el desarrollo rápido (Summerfield, 2010). Es compatible con múltiples plataformas, incluyendo Windows, macOS y Linux, lo que lo hace accesible para una amplia variedad de usuarios (Beazley, 2009).

Gracias a su enfoque en la legibilidad y la simplicidad, Python es una opción popular tanto para principiantes como para desarrolladores experimentados (Zelle, 2016). Su capacidad para integrarse con otros lenguajes y tecnologías, junto con su flexibilidad y potencia, lo convierten en una herramienta valiosa en el desarrollo de software moderno (Raschka, 2015).

## **Java**

Java es un lenguaje de programación de alto nivel, ampliamente utilizado en el desarrollo de aplicaciones empresariales, aplicaciones móviles (especialmente en Android), y sistemas embebidos (Horstmann, 2013). Fue desarrollado originalmente por Sun Microsystems en 1995 y ahora es mantenido por Oracle Corporation (Oracle, 2023). Java es conocido por su principio de

"escribir una vez, ejecutar en cualquier lugar" (WORA, por sus siglas en inglés), lo que significa que el código Java compilado puede ejecutarse en cualquier plataforma que tenga instalada la máquina virtual de Java (JVM) (Gosling et al., 2005).

Java es un lenguaje orientado a objetos, lo que facilita la modularidad y reutilización del código (Bloch, 2018). Su sintaxis es similar a la de C++, pero con una gestión de memoria más sencilla gracias al recolector de basura, que maneja automáticamente la liberación de memoria no utilizada (Arnold et al., 2005). El lenguaje es robusto y seguro, con características que ayudan a prevenir errores comunes en la programación, como el manejo de excepciones y la verificación de tipos en tiempo de compilación (Schildt, 2019). Además, Java tiene una rica biblioteca estándar que proporciona una amplia gama de funcionalidades listas para usar, desde estructuras de datos hasta redes y gráficos (Sierra & Bates, 2015).

Java también es conocido por su gran comunidad de desarrolladores y su ecosistema de herramientas y frameworks, como Spring e Hibernate, que facilitan el desarrollo de aplicaciones complejas (Johnson, 2004). Su versatilidad y estabilidad han hecho de Java una opción popular para proyectos a largo plazo en diversas industrias (Eckel, 2006).

## C#

C# es un lenguaje de programación desarrollado por Microsoft que se ejecuta en el marco de trabajo .NET (Microsoft, 2023a). Es un lenguaje de propósito general, orientado a objetos, que combina la facilidad de uso de lenguajes como Visual Basic con la potencia y flexibilidad de C++ (Troelsen & Japikse, 2021). C# es conocido por su sintaxis clara y su capacidad para desarrollar una amplia variedad de aplicaciones, desde software de escritorio hasta aplicaciones web y servicios en la nube (Albahari & Albahari, 2020).

Una de las características destacadas de C# es su fuerte tipado, lo que ayuda a prevenir errores comunes en la programación al verificar los tipos de datos en tiempo de compilación (Skeet, 2019). Además, C# soporta características modernas como la programación asíncrona, lo que facilita la creación de aplicaciones que pueden manejar múltiples tareas al mismo tiempo de manera eficiente (Wagner, 2021).

C# también se integra de manera nativa con el entorno de desarrollo integrado (IDE) de Microsoft, Visual Studio, que proporciona herramientas avanzadas para el desarrollo, depuración y despliegue de aplicaciones (Microsoft, 2023b). Además, gracias a la plataforma .NET, C# permite el desarrollo de aplicaciones multiplataforma mediante .NET Core, lo que amplía su alcance a sistemas operativos como macOS y Linux (Price, 2020).

## C++

C++ es un lenguaje de programación de propósito general, conocido por su eficiencia y flexibilidad. Desarrollado por Bjarne Stroustrup en 1985 como una extensión del lenguaje C, C++ introduce características de programación orientada a objetos, lo que permite a los desarrolladores crear aplicaciones modulares y reutilizables (Stroustrup, 2013). Este lenguaje es ampliamente utilizado en el desarrollo de software de sistemas, aplicaciones de escritorio, videojuegos, y sistemas embebidos (Lippman et al., 2012).

C++ es un lenguaje compilado, lo que significa que el código fuente se traduce a código máquina antes de ser ejecutado, lo que resulta en un rendimiento optimizado (Meyers, 2014). Además, ofrece un control detallado sobre los recursos del sistema, como la memoria y el procesamiento, lo que lo hace ideal para aplicaciones que requieren alta eficiencia (Prata, 2012).

Una de las características más destacadas de C++ es su capacidad para soportar múltiples paradigmas de programación, incluyendo la programación orientada a objetos, la programación genérica y la programación funcional (Stroustrup, 2014). Esto permite a los desarrolladores elegir el enfoque más adecuado para cada problema específico.

C++ también es conocido por su extensa biblioteca estándar, que proporciona herramientas para realizar tareas comunes como manipulación de cadenas, manejo de archivos y estructuras de datos (Josuttis, 2012). A pesar de su complejidad, C++ sigue siendo un lenguaje popular entre los desarrolladores debido a su potencia y versatilidad.

### **JavaScript**

JavaScript es un lenguaje de programación de alto nivel, dinámico y ampliamente utilizado para el desarrollo web. Originalmente desarrollado por Netscape, JavaScript es el lenguaje de scripting del lado del cliente más popular en la web (Flanagan, 2020). Permite a los desarrolladores crear páginas web interactivas y dinámicas, mejorando la experiencia del usuario al permitir la manipulación de elementos HTML y CSS en tiempo real (Goodman, 2010).

Una de las características más destacadas de JavaScript es su capacidad para ejecutarse en cualquier navegador web moderno, gracias a los intérpretes de JavaScript integrados en navegadores como Google Chrome, Firefox, Safari y Microsoft Edge (Resig & Bibeault, 2013). Esto lo hace altamente accesible y versátil para una amplia variedad de aplicaciones web.

JavaScript es un lenguaje basado en prototipos y soporta programación orientada a objetos, funcional e imperativa (Crockford, 2008). Su sintaxis es relativamente sencilla y se asemeja a C#, lo que facilita su aprendizaje para aquellos familiarizados con otros lenguajes de programación

(Osmani, 2014). Además, JavaScript es compatible con el modelo de Document Object Model (DOM), lo que permite a los desarrolladores interactuar y modificar la estructura de documentos web de manera eficiente (Zakas, 2012).

Con el tiempo, JavaScript ha evolucionado significativamente, incorporando características avanzadas a través de actualizaciones como ECMAScript 6 (ES6) y versiones posteriores (Haverbeke, 2018). Estas actualizaciones han mejorado el lenguaje con nuevas funcionalidades, como clases, módulos y promesas, que facilitan el desarrollo de aplicaciones complejas (Axel, 2016).

### ***3.1.5 Softwares de programación***

Un software de programación es una herramienta que permite a los desarrolladores escribir, probar y depurar código para crear aplicaciones y sistemas informáticos (Sommerville, 2016). Estos programas proporcionan un entorno de desarrollo integrado (IDE) que facilita el proceso de programación al ofrecer características como resaltado de sintaxis, autocompletado de código, y herramientas de depuración. Los IDEs más avanzados también incluyen capacidades para la gestión de proyectos, integración con sistemas de control de versiones, y soporte para múltiples lenguajes de programación (Pressman & Maxim, 2015).

El objetivo principal de un software de programación es mejorar la eficiencia y productividad de los desarrolladores al simplificar tareas complejas y automatizar procesos repetitivos (Gamma et al., 1994). Por ejemplo, herramientas como Visual Studio, Eclipse, y IntelliJ IDEA son populares entre los desarrolladores debido a su capacidad para integrarse con una amplia gama de lenguajes y tecnologías, permitiendo un flujo de trabajo más fluido y organizado (Fowler, 2018).

Además, estos programas suelen ofrecer simuladores y emuladores que permiten a los desarrolladores probar sus aplicaciones en diferentes entornos sin necesidad de hardware adicional. Esto es especialmente útil en el desarrollo de aplicaciones móviles y sistemas embebidos (Meier, 2010).

No todos los programas están diseñados para cumplir un único propósito, ya que sus características y funcionalidades varían según su función específica (Martin, 2008). Esto significa que el número de prestaciones que un programa puede ofrecer depende de su objetivo particular. Algunos de los programas más utilizados en la industria tecnológica incluyen herramientas para el desarrollo de aplicaciones web, gestión de bases de datos, análisis de datos y diseño gráfico. Cada uno de estos programas está optimizado para tareas específicas, lo que permite a los usuarios seleccionar la herramienta más adecuada según sus necesidades particulares (McConnell, 2004). Esta diversidad en las funcionalidades de los programas es esencial para abordar la amplia gama de desafíos que presentan diferentes sectores, como el de la industria petrolera, donde se requiere un manejo eficiente de grandes volúmenes de datos y la capacidad de realizar análisis complejos para optimizar las operaciones y la toma de decisiones estratégicas (Dumbill, 2012), algunos de los más usados son:

- Adobe Dreamweaver
- Lazarus
- Delphi
- Eclipse
- IBM COBOL
- PyCharm

- Netbeans IDE
- Genexus
- CodeLite
- Atom.
- Flutter
- Ionic
- React Native
- Next.js

### **Adobe Dreamweaver**

Adobe Dreamweaver es un software de diseño y desarrollo web desarrollado por Adobe Inc. Este programa es popular entre diseñadores y desarrolladores web debido a su capacidad para facilitar la creación, codificación y gestión de sitios web y contenido móvil (Adobe, 2023). Dreamweaver proporciona un entorno de desarrollo integrado (IDE) que combina una interfaz visual de diseño con un editor de código completo, compatible con lenguajes como HTML, CSS, JavaScript y otros lenguajes de programación web (Smith, 2019).

Una de las características más destacadas de Dreamweaver es su capacidad para alternar entre una vista de diseño visual y una vista de código, lo que permite a los usuarios trabajar en el diseño visual del sitio web mientras tienen acceso directo al código subyacente. Esto hace que Dreamweaver sea accesible tanto para principiantes como para desarrolladores experimentados (Johnson, 2020).

Además, Dreamweaver ofrece herramientas para el diseño responsivo, lo que permite a los diseñadores crear sitios web que se adapten a diferentes tamaños de pantalla y dispositivos.

También incluye funciones de gestión de sitios, facilitando la organización y actualización de archivos en un servidor web (Adobe, 2023).

El software es compatible con varios sistemas de control de versiones, como Git, y se integra con otras herramientas de Adobe Creative Cloud, como Photoshop e Illustrator, permitiendo una colaboración fluida en proyectos de diseño web (Williams, 2018).

### **Lazarus**

Lazarus es un entorno de desarrollo integrado (IDE) de código abierto para el lenguaje de programación Pascal, específicamente diseñado para Free Pascal Compiler (FPC) (Janssen, 2019). Ofrece una interfaz similar a la de Delphi, lo que permite a los desarrolladores crear aplicaciones de escritorio con facilidad (Kahn, 2015). Lazarus es conocido por su capacidad para compilar aplicaciones en múltiples plataformas, incluyendo Windows, macOS, y Linux, sin necesidad de realizar cambios significativos en el código fuente (Smith, 2018).

Una de las características más destacadas de Lazarus es su enfoque en la programación visual. Los desarrolladores pueden diseñar interfaces de usuario utilizando un editor visual de formularios, lo que simplifica el proceso de creación de aplicaciones con interfaces gráficas (Kahn, 2015). Además, Lazarus soporta la programación orientada a objetos, lo que facilita la organización y reutilización del código (Janssen, 2019).

Lazarus también incluye una amplia gama de componentes y bibliotecas que permiten a los desarrolladores añadir funcionalidades avanzadas a sus aplicaciones, como acceso a bases de datos, gráficos, y redes (Smith, 2018). Su comunidad activa contribuye regularmente con nuevas

extensiones y mejoras, lo que garantiza que el entorno se mantenga actualizado con las últimas tecnologías (Lazarus Team, 2023).

### **Delphi**

Delphi es un entorno de desarrollo integrado (IDE) y un lenguaje de programación desarrollado por Embarcadero Technologies. Es conocido por su capacidad para crear aplicaciones de software de manera rápida y eficiente, utilizando el lenguaje de programación Object Pascal (Cantu, 2021). Delphi es particularmente popular para el desarrollo de aplicaciones de escritorio, móviles y bases de datos (Embarcadero Technologies, 2023).

El IDE de Delphi proporciona un conjunto completo de herramientas para el diseño visual de interfaces de usuario, lo que permite a los desarrolladores arrastrar y soltar componentes en un formulario para crear aplicaciones con interfaces gráficas de usuario (GUI) de manera intuitiva (Teti, 2019). Esto facilita el desarrollo rápido de aplicaciones (RAD, por sus siglas en inglés)..

Delphi es compatible con múltiples plataformas, incluyendo Windows, macOS, iOS y Android, lo que permite a los desarrolladores crear aplicaciones multiplataforma con un solo código base (Cantu, 2021). Además, ofrece una amplia gama de bibliotecas y componentes que facilitan la integración con bases de datos, servicios web y otras tecnologías (Embarcadero Technologies, 2023).

Una de las características destacadas de Delphi es su capacidad para compilar aplicaciones nativas de alto rendimiento, lo que lo hace ideal para aplicaciones que requieren un uso intensivo de recursos (Teti, 2019). También es conocido por su estabilidad y la facilidad con la que se pueden mantener y escalar las aplicaciones desarrolladas (Cantu, 2021).

## **Eclipse**

Eclipse es un entorno de desarrollo integrado (IDE) de código abierto, ampliamente utilizado por desarrolladores para crear aplicaciones en una variedad de lenguajes de programación, incluidos Java, C++, Python, y más (Eclipse Foundation, 2023). Originalmente desarrollado por IBM, Eclipse es ahora mantenido por la Fundación Eclipse y es conocido por su extensibilidad y su arquitectura basada en plugins (Holzner, 2004).

Una de las características más destacadas de Eclipse es su capacidad para ser altamente personalizable. Los desarrolladores pueden ampliar sus funcionalidades mediante la instalación de plugins adicionales, lo que permite adaptar el IDE a las necesidades específicas de un proyecto o lenguaje de programación (Vogel, 2015). Esto lo hace especialmente popular en el desarrollo de aplicaciones Java, donde ofrece herramientas avanzadas para la depuración, pruebas, y gestión de proyectos (Burnette, 2005).

Eclipse también soporta el desarrollo de aplicaciones móviles, web, y empresariales, gracias a su amplia gama de herramientas y plugins disponibles (Clayberg & Rubel, 2008). Su interfaz de usuario es intuitiva y ofrece características como resaltado de sintaxis, autocompletado de código, y refactorización, lo que facilita el proceso de desarrollo (Gallardo et al., 2012).

## **IBM COBOL**

IBM COBOL es un entorno de desarrollo para el lenguaje de programación COBOL, ampliamente utilizado en aplicaciones empresariales y sistemas de misión crítica. COBOL, que significa Common Business-Oriented Language, es uno de los lenguajes de programación más antiguos y sigue siendo relevante en la actualidad, especialmente en sectores como la banca, seguros y administración pública (Stern & Stern, 2014). El software IBM COBOL proporciona un

conjunto de herramientas para el desarrollo, depuración y mantenimiento de aplicaciones COBOL. Ofrece características avanzadas como la integración con bases de datos, soporte para servicios web y capacidades de modernización de aplicaciones, lo que permite a las empresas actualizar y optimizar sus sistemas legados (IBM, 2023).

Una de las ventajas clave de IBM COBOL es su capacidad para manejar grandes volúmenes de transacciones y datos, lo que lo hace ideal para aplicaciones que requieren alta fiabilidad y eficiencia (Murach, 2013). Además, el entorno de desarrollo está diseñado para facilitar la colaboración entre equipos de desarrollo, permitiendo a los programadores trabajar en proyectos complejos de manera efectiva (IBM, 2023).

### **PyCharm**

PyCharm es un entorno de desarrollo integrado (IDE) desarrollado por JetBrains, diseñado específicamente para el lenguaje de programación Python (JetBrains, 2023). Es ampliamente utilizado por desarrolladores debido a su potente conjunto de herramientas que facilitan la escritura, depuración y prueba de código Python. PyCharm ofrece características avanzadas como autocompletado de código, resaltado de sintaxis, y refactorización, lo que mejora la productividad y eficiencia de los programadores (Roche, 2018).

Una de las principales ventajas de PyCharm es su capacidad para integrarse con sistemas de control de versiones como Git, así como con herramientas de desarrollo web y frameworks populares como Django y Flask (Percival, 2017). Esto lo convierte en una opción ideal para el desarrollo de aplicaciones web y proyectos de ciencia de datos.

PyCharm también incluye un depurador visual que permite a los desarrolladores inspeccionar el flujo de ejecución del programa y corregir errores de manera más eficiente (Sweigart, 2020). Además, ofrece soporte para pruebas unitarias, facilitando la creación de código robusto y confiable (Summerfield, 2019).

### **Netbeans IDE**

NetBeans IDE es un entorno de desarrollo integrado (IDE) de código abierto, ampliamente utilizado para el desarrollo de aplicaciones en una variedad de lenguajes de programación, incluyendo Java, PHP, C++, y HTML5 (Apache Software Foundation, 2023). Desarrollado originalmente por Sun Microsystems y ahora mantenido por la comunidad Apache, NetBeans es conocido por su facilidad de uso y su capacidad para mejorar la productividad de los desarrolladores (Boudreau et al., 2015).

NetBeans ofrece un conjunto completo de herramientas que facilitan el desarrollo de software, como el resaltado de sintaxis, autocompletado de código, y un depurador integrado (Wielenga, 2015). Además, proporciona un editor visual para el diseño de interfaces gráficas de usuario, lo que simplifica el proceso de creación de aplicaciones con interfaces atractivas (Petri, 2013).

Una de las características destacadas de NetBeans es su modularidad. Los desarrolladores pueden ampliar sus funcionalidades mediante la instalación de módulos adicionales, lo que permite personalizar el IDE según las necesidades específicas del proyecto (Böck, 2011). Además, NetBeans soporta el desarrollo de aplicaciones empresariales mediante su integración con servidores de aplicaciones como GlassFish y Apache Tomcat (Heffelfinger, 2015).

NetBeans también es compatible con sistemas de control de versiones como Git, Mercurial y Subversion, lo que facilita la gestión de proyectos en equipo (Salter & Dantas, 2014). Su capacidad para desarrollar aplicaciones multiplataforma y su soporte para las últimas tecnologías de desarrollo web lo convierten en una herramienta versátil y poderosa para desarrolladores de todo el mundo (Wielenga, 2015).

### **GeneXus**

GeneXus es una herramienta de desarrollo de software que permite a los desarrolladores crear aplicaciones multiplataforma de manera eficiente. Diseñado por la empresa uruguaya Artech, GeneXus se destaca por su enfoque en la automatización del desarrollo de software, generando automáticamente código para diferentes plataformas a partir de un modelo de datos y reglas de negocio definidos por el usuario (Artech, 2023). Una de las características más notables de GeneXus es su capacidad para generar aplicaciones para múltiples entornos, incluyendo web, móviles y de escritorio, sin necesidad de escribir código específico para cada plataforma. Esto se logra mediante la utilización de un lenguaje de alto nivel que describe la lógica de negocio y los datos, a partir del cual GeneXus genera el código necesario en lenguajes como Java, C#, y JavaScript, entre otros (Artech, 2023).

GeneXus también ofrece integración con bases de datos, permitiendo a los desarrolladores definir y gestionar esquemas de bases de datos de manera visual. Además, incluye herramientas para la gestión del ciclo de vida del software, facilitando la implementación continua y el mantenimiento de las aplicaciones desarrolladas (Artech, 2023).

### **CodeLite**

CodeLite es un entorno de desarrollo integrado (IDE) de código abierto diseñado principalmente para el desarrollo de aplicaciones en lenguajes como C, C++, PHP, y JavaScript (CodeLite, 2023). Es conocido por su ligereza y eficiencia, lo que lo convierte en una opción popular entre los desarrolladores que buscan una herramienta rápida y funcional para escribir y depurar código (Eran, 2019).

Una de las características más destacadas de CodeLite es su capacidad para integrarse con múltiples compiladores, lo que permite a los desarrolladores elegir el compilador que mejor se adapte a sus necesidades (CodeLite, 2023). Además, ofrece soporte para la depuración de código, resaltado de sintaxis, y autocompletado, lo que facilita el proceso de desarrollo al proporcionar un entorno de programación intuitivo (Eran, 2019).

CodeLite también incluye herramientas para la gestión de proyectos, permitiendo a los desarrolladores organizar y navegar fácilmente por sus proyectos de software (CodeLite, 2023). Su interfaz es altamente personalizable, lo que permite a los usuarios adaptar el entorno de trabajo a sus preferencias personales (Eran, 2019).

### **Atom**

Atom es un editor de texto de código abierto desarrollado por GitHub, diseñado para ser altamente personalizable y extensible (GitHub, 2023). Es popular entre desarrolladores debido a su interfaz amigable y su capacidad para soportar múltiples lenguajes de programación. Atom permite a los usuarios modificar su entorno de trabajo mediante la instalación de paquetes y temas, lo que facilita la adaptación del editor a las necesidades específicas de cada proyecto (Hennessy, 2016).

Una de las características más destacadas de Atom es su integración con Git y GitHub, lo que permite a los desarrolladores gestionar el control de versiones directamente desde el editor (Flight, 2018). Además, Atom ofrece características avanzadas como autocompletado de código, resaltado de sintaxis, y un navegador de archivos integrado, lo que mejora la productividad al facilitar la navegación y edición de proyectos complejos (Atom, 2023).

Atom también soporta la colaboración en tiempo real a través de Teletype, una extensión que permite a los desarrolladores trabajar juntos en el mismo código desde diferentes ubicaciones (GitHub, 2022). Su arquitectura basada en Electron permite que Atom sea multiplataforma, funcionando en sistemas operativos como Windows, macOS y Linux (Electron, 2023).

### **Flutter**

Flutter es un kit de desarrollo de software (SDK) de código abierto creado por Google, diseñado para facilitar la creación de aplicaciones nativas de alta calidad para dispositivos móviles, web y de escritorio desde una única base de código (Google, 2023a). Lanzado en 2018, Flutter se ha destacado por su capacidad para desarrollar interfaces de usuario expresivas y flexibles gracias a su arquitectura basada en widgets (Napoli, 2020).

Una de las características más notables de Flutter es su motor de renderizado rápido, que utiliza la biblioteca gráfica Skia para ofrecer animaciones suaves y transiciones de interfaz de usuario a 60 cuadros por segundo (Windmill, 2021). Esto permite a los desarrolladores crear aplicaciones con un rendimiento cercano al nativo en múltiples plataformas.

Flutter utiliza el lenguaje de programación Dart, también desarrollado por Google, que es conocido por su facilidad de aprendizaje y su capacidad para compilar tanto en código nativo como

en JavaScript (Google, 2023b). Esto permite que las aplicaciones Flutter se ejecuten eficientemente en dispositivos iOS y Android, así como en navegadores web.

El enfoque de Flutter en el desarrollo de interfaces de usuario se ve reforzado por su amplia colección de widgets personalizables, que permiten a los desarrolladores construir interfaces complejas de manera sencilla (Biessek, 2019). Además, Flutter proporciona herramientas como "hot reload", que permite ver los cambios en el código de manera instantánea sin perder el estado de la aplicación, lo que acelera el proceso de desarrollo (Sande & Galloway, 2021).

### **Ionic**

Ionic es un framework de desarrollo de código abierto que permite a los desarrolladores crear aplicaciones móviles híbridas y aplicaciones web progresivas (PWA) utilizando tecnologías web como HTML, CSS y JavaScript (Ionic, 2023). Desarrollado por Drifty Co., Ionic se basa en Angular y Apache Cordova, lo que permite a los desarrolladores construir aplicaciones que se ejecutan en múltiples plataformas, incluyendo iOS, Android y la web, a partir de un único código base (Holmes, 2019).

Una de las características más destacadas de Ionic es su extensa biblioteca de componentes de interfaz de usuario predefinidos, que permite a los desarrolladores crear aplicaciones con una apariencia nativa y atractiva (Buchanan, 2020). Estos componentes están optimizados para el rendimiento en dispositivos móviles y se pueden personalizar fácilmente para adaptarse a las necesidades específicas de cada proyecto.

Ionic también ofrece herramientas de desarrollo como Ionic CLI, que facilita la creación, construcción y despliegue de aplicaciones, así como Capacitor, una plataforma que permite

acceder a las API nativas del dispositivo y extender las capacidades de las aplicaciones híbridas (Ionic, 2023).

### **React Native**

React Native es un marco de desarrollo de código abierto creado por Facebook que permite a los desarrolladores construir aplicaciones móviles nativas utilizando JavaScript y React (Facebook, 2023). Lanzado en 2015, React Native se ha convertido en una herramienta popular para el desarrollo de aplicaciones móviles debido a su capacidad para crear aplicaciones que funcionan en sistemas operativos iOS y Android desde una única base de código (Eisenman, 2015).

Una de las características más destacadas de React Native es su uso de componentes nativos, lo que permite que las aplicaciones tengan un rendimiento similar al de las aplicaciones nativas tradicionales (Wixom & Gao, 2015). A diferencia de las aplicaciones híbridas que utilizan tecnologías web, React Native traduce sus componentes a vistas nativas, lo que mejora la experiencia del usuario y el rendimiento de la aplicación (Bonnie, 2017).

React Native utiliza el mismo diseño de interfaz de usuario que React, lo que permite a los desarrolladores construir interfaces de usuario ricas y dinámicas (Novick, 2017). Además, ofrece herramientas como "hot reloading", que permite ver los cambios en el código en tiempo real sin necesidad de recompilar la aplicación, lo que acelera el proceso de desarrollo (Dabit, 2019).

### **Next.js**

Next.js es un marco de desarrollo de código abierto basado en React, diseñado para crear aplicaciones web y sitios web estáticos de manera eficiente (Vercel, 2023). Desarrollado por Vercel, Next.js se destaca por su capacidad para ofrecer renderizado del lado del servidor (SSR),

lo que mejora el rendimiento y la optimización para motores de búsqueda (SEO) al generar contenido HTML en el servidor antes de enviarlo al cliente (Lynch, 2021). Este enfoque permite que las aplicaciones carguen más rápido y sean más accesibles a los motores de búsqueda.

Una de las características más notables de Next.js es su simplicidad en la configuración y su capacidad para manejar enrutamiento automático basado en el sistema de archivos, lo que facilita la creación de rutas dinámicas y estáticas sin necesidad de configuraciones complejas (Bucă, 2020). Además, Next.js soporta la generación de sitios estáticos (SSG) y la renderización híbrida, permitiendo a los desarrolladores elegir el método de renderización más adecuado para cada página de su aplicación (Roberts, 2020).

Next.js también ofrece una experiencia de desarrollo mejorada con características como el recargo en caliente, la división automática de código y la optimización de imágenes, lo que permite a los desarrolladores construir aplicaciones modernas y de alto rendimiento con facilidad (Vercel, 2023). Su integración con el ecosistema de React y su compatibilidad con herramientas modernas de desarrollo lo convierten en una opción popular para proyectos que requieren rendimiento, flexibilidad y escalabilidad (Lynch, 2021).

### ***3.1.6 Visualización de datos***

La visualización de datos es el proceso de representar datos en un formato visual, como gráficos, diagramas o mapas, para facilitar su comprensión y análisis (Few, 2009). Este enfoque permite a los usuarios identificar patrones, tendencias y anomalías en grandes conjuntos de datos de manera más intuitiva que al examinar datos en bruto (Tufte, 2001). La visualización de datos es una herramienta crucial en la toma de decisiones, ya que proporciona una manera clara y concisa de comunicar información compleja (Cairo, 2013).

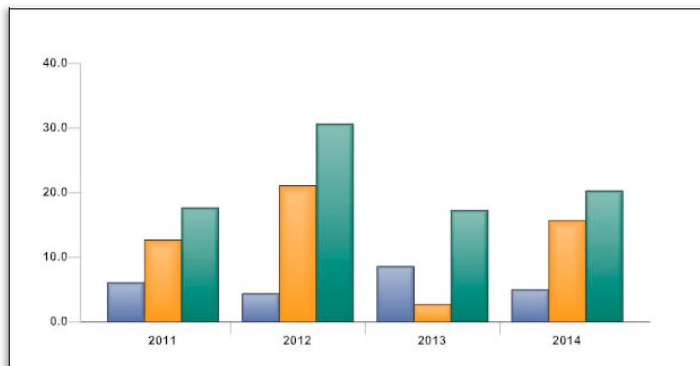
Al utilizar técnicas de visualización, las organizaciones pueden transformar datos abstractos en representaciones visuales que revelan relaciones y estructuras subyacentes (Munzner, 2014). Esto es especialmente útil en campos como la ciencia de datos, el análisis de negocios y la investigación científica, donde la capacidad de interpretar y comunicar hallazgos de manera efectiva es esencial (Ware, 2012).

Las herramientas de visualización de datos, como Tableau, Power BI y D3.js, ofrecen una variedad de opciones para crear visualizaciones interactivas y personalizadas (Murray, 2017). Estas herramientas permiten a los usuarios explorar datos en tiempo real, ajustar parámetros y obtener insights inmediatos (Kirk, 2016).

### ***Gráfico de barras***

Un gráfico de barras es un tipo de representación visual que utiliza barras rectangulares para representar datos. Cada barra representa una categoría de datos, y la longitud de la barra es proporcional al valor que representa (Few, 2009). Los gráficos de barras son ideales para comparar cantidades entre diferentes grupos o categorías y se utilizan comúnmente en análisis de ventas, encuestas de opinión y cualquier situación donde se necesite comparar datos categóricos (Kirk, 2016).

**Figura 1** – Ejemplo de gráfico de barras

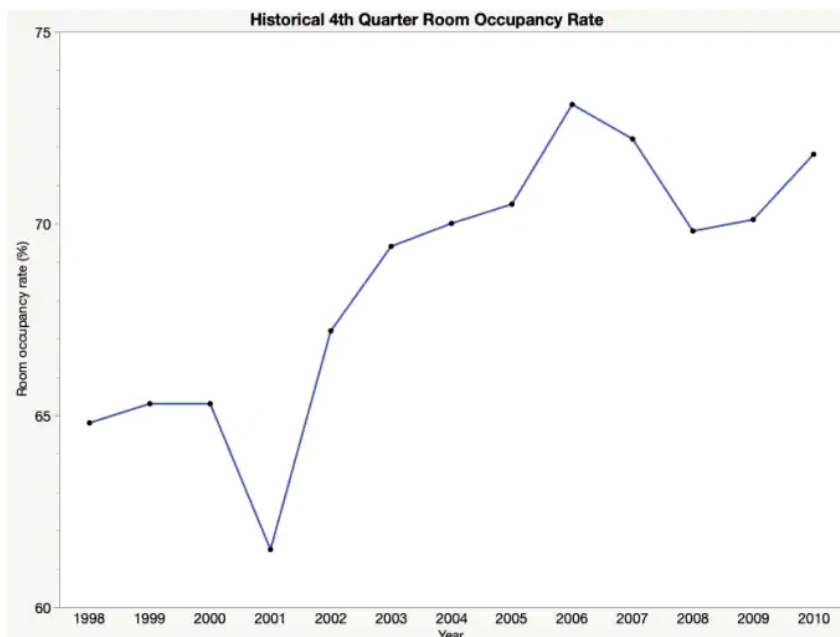


*Tomado de Socialmediapymes (Gráficos de barras, cuándo usarlos y cómo diseñarlos), por C. D. Soloaga, 2020.*

### **Gráfico de líneas**

Los gráficos de líneas conectan puntos de datos con líneas y son útiles para mostrar tendencias a lo largo del tiempo (Few, 2009). Son ampliamente utilizados en análisis financieros, meteorología, y cualquier campo donde se necesite observar cambios en los datos a través de un período, como el crecimiento de ventas mensuales o las fluctuaciones de temperatura (Tufte, 2001). Estos gráficos son particularmente efectivos para visualizar datos continuos y para identificar patrones y tendencias en conjuntos de datos temporales (Cairo, 2013).

**Figura 2** – Ejemplo de gráfico de líneas

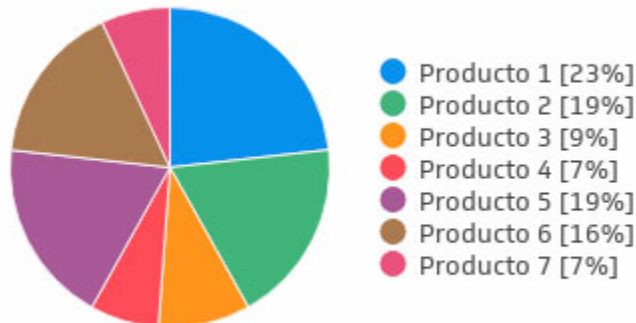


Tomado de JMP Statistical Discovery (Gráfico de líneas).

### **Gráfico de pastel**

También conocido como gráfico circular, este tipo de gráfico divide un círculo en sectores que representan proporciones de un todo (Few, 2007). Es útil para mostrar la composición porcentual de un conjunto de datos (Tufté, 2001). Los gráficos de pastel se utilizan frecuentemente en informes de distribución de mercado o para mostrar la participación de mercado de diferentes productos (Cairo, 2013). Sin embargo, algunos expertos en visualización de datos argumentan que los gráficos de barras o columnas pueden ser más efectivos para comparar proporciones (Knaflic, 2015).

**Figura 3** – Ejemplo de gráfico de pastel

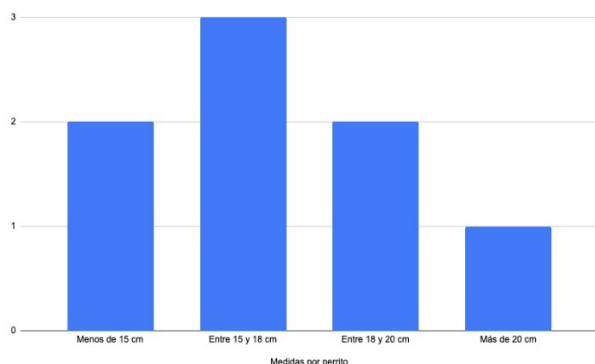


*Tomado de TuDashboard (Gráfica de Pastel. Conoce sus características y cuándo utilizarla), 2020.*

### ***Histograma***

Similar a un gráfico de barras, un histograma se utiliza para representar la distribución de un conjunto de datos continuos. Los histogramas son útiles para mostrar la frecuencia de datos dentro de ciertos rangos y son comúnmente utilizados en estadística para analizar distribuciones de datos, como la distribución de edades o la altura de una población (Spence, 2005). Este tipo de gráfico es esencial para identificar patrones y tendencias en conjuntos de datos grandes, permitiendo a los analistas visualizar cómo se distribuyen los datos en diferentes intervalos (Everitt & Skronidal, 2010).

**Figura 4 – Ejemplo de gráfico histograma**

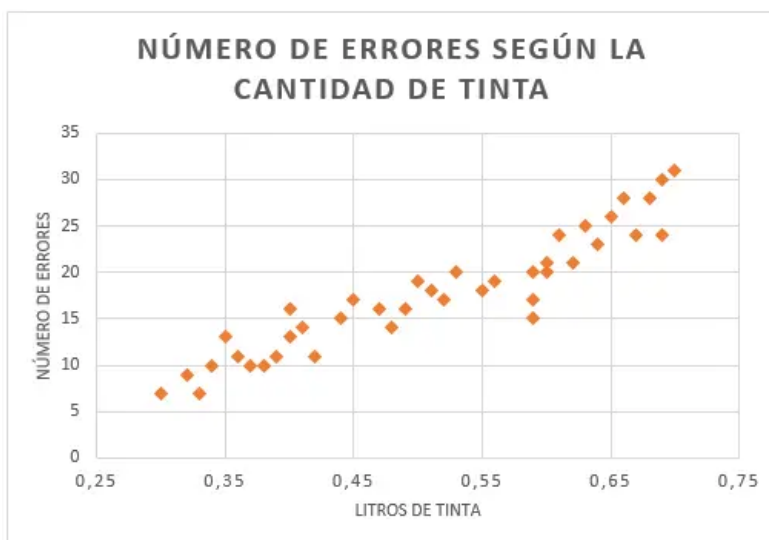


*Tomado de Concepto (Histograma), por Editorial Etecé, 2024.*

**Gráfico de Dispersión**

Este gráfico utiliza puntos para representar valores de dos variables diferentes, permitiendo observar la relación entre ellas (Friendly & Denis, 2005). Los gráficos de dispersión son útiles para identificar correlaciones o patrones en los datos, y se utilizan frecuentemente en análisis de regresión y estudios científicos (Tufté, 2001). Estos gráficos son particularmente efectivos para visualizar la fuerza y dirección de las relaciones entre variables continuas, facilitando la detección de tendencias, agrupaciones y valores atípicos en conjuntos de datos complejos (Cleveland, 1993).

**Figura 5** – Ejemplo de grafico de dispersión

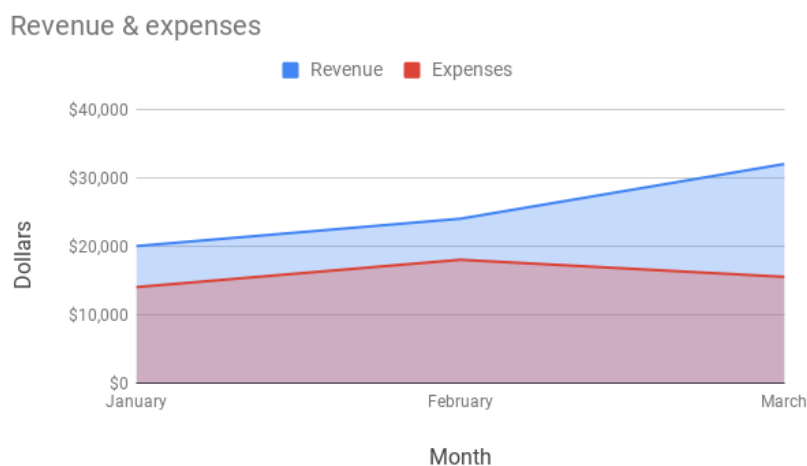


Tomado de Ingenio Empresa (Cómo hacer un diagrama de dispersión: Ejemplo en calidad.), por D. F. Betancourt, 2016.

**Gráfico de Área**

Similar a un gráfico de líneas, pero el área bajo la línea está rellena, lo que enfatiza la magnitud del cambio a lo largo del tiempo (Few, 2009). Se utilizan para mostrar cómo una cantidad cambia en relación con otra, como el ingreso acumulado a lo largo del tiempo (Tuftte, 2001). Los gráficos de área son particularmente efectivos para visualizar tendencias acumulativas y comparar múltiples series de datos a lo largo del tiempo (Cairo, 2016).

**Figura 6 – Ejemplo de gráfico de área**

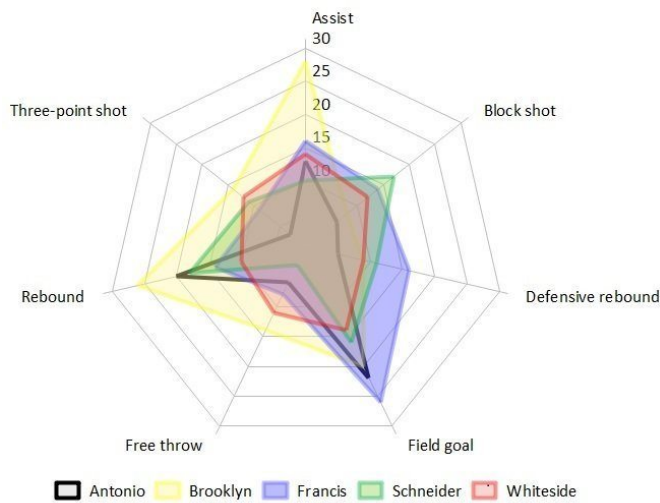


Tomado de Support Google (Gráficos de áreas.), por Google.

***Gráfico de Radar***

También conocido como gráfico de araña, el gráfico de radar se utiliza para mostrar multivariables en un gráfico bidimensional. Cada variable tiene su propio eje que emana desde el centro, lo que permite comparar múltiples variables al mismo tiempo (Friendly, 2009). Es útil para evaluar el rendimiento de un producto en diferentes criterios, proporcionando una representación visual clara de las fortalezas y debilidades relativas (Spence, 2005).

**Figura 7 – Ejemplo de gráfico de radar o araña**

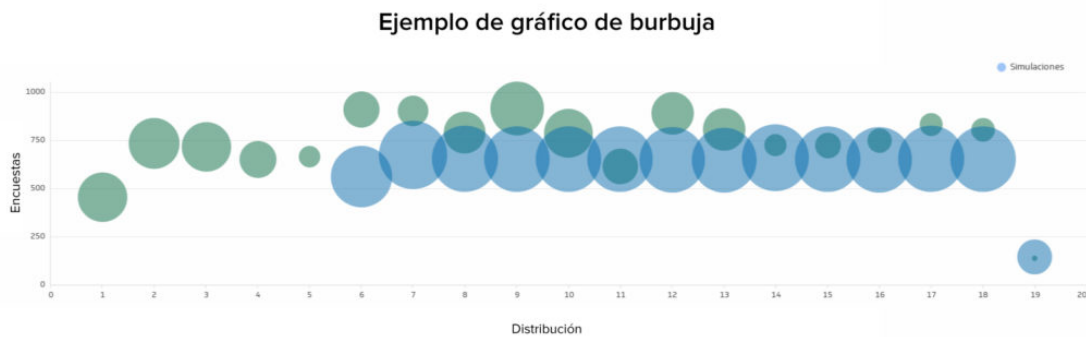


Tomado de edraw Wondershare (Qué es un gráfico de radar y ejemplos).

### ***Gráfico de Burbuja***

Una extensión del gráfico de dispersión, donde el tamaño de las burbujas representa una tercera variable (Few, 2009). Es útil para mostrar la relación entre tres variables y se utiliza en análisis de negocios para evaluar el impacto de diferentes factores en un resultado (Tufté, 2001). Los gráficos de burbuja son particularmente efectivos para visualizar datos multidimensionales en un espacio bidimensional, permitiendo a los analistas identificar patrones y correlaciones complejas entre múltiples variables simultáneamente (Cairo, 2016).

**Figura 8 – Ejemplo de gráfico de burbuja**



Tomado de TuDashboard (Gráfico de burbuja: Características y usos más comunes), 2021.

### 3.2 Capítulo 2 – Selección de Herramientas y Base de Datos

En este capítulo, se aborda la elección estratégica de las tecnologías y recursos que se utilizarán para desarrollar una aplicación web destinada al almacenamiento y análisis de datos en la industria petrolera. La selección de herramientas adecuadas es crucial para garantizar la eficiencia, escalabilidad y mantenibilidad del proyecto (Pressman & Maxim, 2020). Esta decisión implica considerar diversos factores como el rendimiento, la compatibilidad con los sistemas existentes, y la capacidad de manejar grandes volúmenes de datos característicos del sector petrolero (Mohammadpoor & Torabi, 2020). Además, la elección de tecnologías debe tener en cuenta las tendencias actuales en desarrollo web y análisis de datos, asegurando que la aplicación pueda evolucionar con las necesidades cambiantes de la industria (Leau et al., 2012)..

#### 3.2.1 Lenguaje de Programación

JavaScript ha sido elegido como el lenguaje de programación principal debido a su flexibilidad y su amplia adopción en el desarrollo web moderno, lo que permite una integración fluida con las herramientas seleccionadas.

JavaScript es un lenguaje de programación interpretado y de alto nivel, que se ha convertido en una de las piedras angulares del desarrollo web moderno. Originalmente diseñado para añadir interactividad a las páginas web, JavaScript ha evolucionado significativamente y ahora se utiliza tanto en el lado del cliente como en el lado del servidor, gracias a plataformas como Node.js (Flanagan, 2020). Su versatilidad y capacidad para integrarse con HTML y CSS lo convierten en una opción ideal para desarrollar aplicaciones web interactivas y dinámicas.

Una de las razones por las que JavaScript es una elección ideal para la tesis sobre el desarrollo de una aplicación web para el almacenamiento y análisis de datos en la industria petrolera es su compatibilidad con Next.js, un framework basado en React que facilita el renderizado del lado del servidor y la generación de sitios estáticos (Wexler, 2020). JavaScript, en combinación con Next.js, permite crear aplicaciones web rápidas y optimizadas para motores de búsqueda, lo cual es crucial para manejar grandes volúmenes de datos de manera eficiente.

Además, JavaScript es compatible con MongoDB, una base de datos NoSQL que utiliza un modelo de datos basado en documentos (Chodorow, 2013). Esto permite manejar datos no estructurados y semiestructurados de manera flexible, lo cual es común en el sector petrolero. La capacidad de JavaScript para interactuar con bases de datos como MongoDB facilita el desarrollo de aplicaciones que requieren un manejo eficiente de grandes cantidades de datos.

Comparado con otros lenguajes de programación, JavaScript ofrece una curva de aprendizaje relativamente baja y una amplia comunidad de desarrolladores, lo que proporciona acceso a una gran cantidad de recursos y bibliotecas (Simpson, 2015). Esto permite a los desarrolladores implementar soluciones rápidas y efectivas, optimizando el tiempo de desarrollo y reduciendo costos.

### ***3.2.2 Normalizador y Persistencia de Datos***

En cuanto a la gestión de datos, se utilizará MongoDB, una base de datos NoSQL que permite almacenar datos en un formato flexible y escalable, ideal para manejar grandes volúmenes de información no estructurada, característica común en la industria petrolera (MongoDB, 2023).

MongoDB es un sistema de gestión de bases de datos NoSQL de código abierto que utiliza un modelo de datos basado en documentos. A diferencia de los sistemas de bases de datos relacionales tradicionales, MongoDB almacena datos en documentos JSON flexibles, lo que permite una estructura de datos más dinámica y adaptable (Chodorow, 2013). Esta flexibilidad es especialmente útil para aplicaciones que requieren escalabilidad horizontal y necesitan manejar grandes volúmenes de datos no estructurados o semiestructurados, como es común en la industria petrolera.

Una de las principales ventajas de MongoDB es su capacidad para escalar de manera eficiente mediante la distribución de datos en múltiples servidores, garantizando alta disponibilidad y tolerancia a fallos (Banker, 2011). Además, ofrece potentes capacidades de consulta y agregación, permitiendo realizar análisis complejos de datos de manera sencilla. MongoDB es compatible con diversas plataformas y lenguajes de programación, facilitando su integración en entornos de desarrollo modernos (MongoDB, 2023).

La implementación de MongoDB en la tesis es ideal en comparación con otros sistemas de gestión de bases de datos debido a su flexibilidad y escalabilidad. En el contexto de la industria petrolera, donde los datos pueden ser altamente variados y cambiar con frecuencia, la capacidad de MongoDB para manejar datos no estructurados y su arquitectura distribuida lo convierten en una opción superior (Plugge et al., 2015). Además, su modelo de datos basado en documentos

permite una rápida iteración y adaptación a los cambios en los requisitos de datos, lo que es crucial para mantener la competitividad en un sector tan dinámico (Dayley, 2014).

### **3.2.3 Base de Datos**

Además, se integrará la "BP Statistical Review 2022", una fuente confiable de datos energéticos globales, que proporcionará un contexto valioso y datos históricos para el análisis.

La BP Statistical Review of World Energy es una fuente de datos ampliamente reconocida en el sector energético, que ofrece un análisis detallado de las tendencias globales en el consumo y producción de energía. Publicada anualmente, esta revisión proporciona estadísticas sobre diversas fuentes de energía, incluyendo petróleo, gas natural, carbón, energía renovable y nuclear, así como datos sobre emisiones de carbono.

La implementación de la BP Statistical Review 2022 en una tesis sobre el almacenamiento y análisis de datos en la industria petrolera es ideal debido a varias razones:

- **Datos Actualizados y Confiables:** La revisión proporciona datos actualizados y confiables que son esenciales para el análisis de tendencias actuales y futuras en el sector energético (Smil, 2017).
- **Cobertura Global:** Ofrece una perspectiva global del mercado energético, lo que es crucial para entender las dinámicas internacionales que afectan la industria petrolera (Yergin, 2020).
- **Diversidad de Datos:** Incluye una amplia gama de datos sobre diferentes fuentes de energía y su uso, lo que permite realizar análisis comparativos y evaluar la transición energética hacia fuentes más sostenibles (IEA, 2021).

- Contexto Histórico: La revisión proporciona datos históricos que son útiles para identificar tendencias a largo plazo y realizar análisis predictivos (Sovacool et al., 2016).

En comparación con otras bases de datos, la BP Statistical Review es altamente valorada por su exhaustividad y precisión, lo que la convierte en una herramienta esencial para cualquier investigación seria en el ámbito energético. Su uso en la tesis garantiza que el análisis esté respaldado por datos de alta calidad y reconocidos internacionalmente.

### ***3.2.4 Construcción de la interfaz***

Para la construcción de la interfaz de usuario y la lógica del cliente, se ha optado por Next.js, un potente framework de React que ofrece capacidades de renderizado del lado del servidor, lo que mejora el rendimiento y la optimización para motores de búsqueda (Vercel, 2023).

Next.js es un framework de desarrollo de código abierto basado en React, diseñado para facilitar la creación de aplicaciones web y sitios web estáticos de manera eficiente. Desarrollado por Vercel, Next.js se destaca por su capacidad para ofrecer renderizado del lado del servidor (SSR), lo que mejora el rendimiento y la optimización para motores de búsqueda (SEO) al generar contenido HTML en el servidor antes de enviarlo al cliente (Wexler, 2021). Este enfoque permite que las aplicaciones carguen más rápido y sean más accesibles a los motores de búsqueda.

Una de las características más notables de Next.js es su simplicidad en la configuración y su capacidad para manejar enrutamiento automático basado en el sistema de archivos, lo que facilita la creación de rutas dinámicas y estáticas sin necesidad de configuraciones complejas (Papp, 2020). Además, Next.js soporta la generación de sitios estáticos (SSG) y la renderización

híbrida, permitiendo a los desarrolladores elegir el método de renderización más adecuado para cada página de su aplicación (Vercel, 2023).

La implementación de Next.js en la tesis es ideal en comparación con otras aplicaciones debido a su flexibilidad y capacidad para manejar aplicaciones web modernas que requieren un alto rendimiento y una excelente experiencia de usuario. Next.js permite a los desarrolladores construir aplicaciones escalables y mantenibles, con soporte para características avanzadas como la división automática de código y la optimización de imágenes (Kiely, 2020). Además, su integración con el ecosistema de React y su compatibilidad con herramientas modernas de desarrollo lo convierten en una opción popular para proyectos que requieren rendimiento, flexibilidad y escalabilidad (Wexler, 2021).

En el contexto de la tesis, donde el objetivo es desarrollar una aplicación web para el almacenamiento y análisis de datos en la industria petrolera, Next.js ofrece las herramientas necesarias para crear una interfaz de usuario eficiente y optimizada, capaz de manejar grandes volúmenes de datos y proporcionar una experiencia de usuario fluida. Su capacidad para integrar fácilmente con otras tecnologías y su enfoque en el desarrollo rápido y eficiente hacen de Next.js la elección ideal para este proyecto (Papp, 2020).

### ***3.2.5 Entorno de Desarrollo Integrado***

Visual Studio Code se ha elegido como el entorno de desarrollo integrado (IDE) por su versatilidad y amplia gama de extensiones que facilitan la escritura y depuración del código.

Visual Studio Code (VS Code) es un entorno de desarrollo integrado (IDE) altamente popular y de código abierto, desarrollado por Microsoft. Se destaca por su ligereza, velocidad y

extensibilidad, lo que lo convierte en una herramienta ideal para desarrolladores que trabajan en una amplia gama de lenguajes de programación, incluyendo JavaScript, Python, C++, y más (Pecanac, 2021). VS Code ofrece una experiencia de desarrollo enriquecida con características como autocompletado de código, depuración integrada, control de versiones, y una terminal incorporada (Johnson, 2022). Su arquitectura basada en extensiones permite a los desarrolladores personalizar el entorno de trabajo según sus necesidades específicas, instalando extensiones que añaden funcionalidades adicionales, como soporte para nuevos lenguajes o herramientas de desarrollo (Smyth, 2020).

La implementación de Visual Studio Code en la tesis es ideal debido a varias razones. Primero, su integración con Git y otros sistemas de control de versiones facilita la gestión del código fuente, lo cual es crucial para mantener un historial claro de los cambios y colaboraciones en el proyecto (Pecanac, 2021). Además, su soporte para JavaScript y tecnologías relacionadas, como Node.js y React, lo hace especialmente adecuado para el desarrollo de aplicaciones web modernas, como la que se propone en la tesis (Johnson, 2022).

En comparación con otros IDEs, VS Code es conocido por su rendimiento rápido y su capacidad para manejar proyectos de gran escala sin comprometer la velocidad (Smyth, 2020). Su comunidad activa y su ecosistema de extensiones en constante crecimiento aseguran que los desarrolladores tengan acceso a las últimas herramientas y tecnologías (Microsoft, 2023). En el contexto de la tesis, donde se busca desarrollar una aplicación web eficiente para el almacenamiento y análisis de datos en la industria petrolera, Visual Studio Code proporciona un entorno de desarrollo flexible y potente que facilita la implementación de soluciones innovadoras y adaptables.

### ***3.2.6 Navegador de Preferencia***

Para probar y desplegar la aplicación, Google Chrome será el navegador de preferencia, aprovechando sus herramientas de desarrollo avanzadas para depurar y optimizar el rendimiento de la aplicación (Google, 2023a). La combinación de estas herramientas y tecnologías está diseñada para proporcionar una solución robusta y eficiente, capaz de satisfacer las complejas necesidades de análisis de datos en el sector petrolero.

Google Chrome es un navegador web desarrollado por Google, conocido por su velocidad, simplicidad y eficiencia. Desde su lanzamiento en 2008, Chrome ha ganado popularidad rápidamente y se ha convertido en uno de los navegadores más utilizados en el mundo (StatCounter, 2023). Chrome se basa en el motor de renderizado Blink y el motor JavaScript V8, lo que le permite cargar páginas web rápidamente y ejecutar aplicaciones web de manera eficiente (Google, 2023b). Además, Chrome ofrece una interfaz de usuario minimalista que maximiza el espacio de visualización de las páginas web, y su barra de direcciones, conocida como Omnibox, permite realizar búsquedas y navegar de manera integrada (Garsiel & Irish, 2011).

Una de las razones por las que la implementación de Google Chrome en la tesis es ideal en comparación con otras aplicaciones es su conjunto de herramientas de desarrollo avanzadas. Las Chrome DevTools proporcionan un entorno robusto para depurar y optimizar aplicaciones web, permitiendo a los desarrolladores inspeccionar elementos, monitorear el rendimiento, y probar la compatibilidad de sus aplicaciones en tiempo real (Google Developers, 2023). Esto es esencial para asegurar que la aplicación web desarrollada en la tesis funcione de manera óptima y sin errores.

Además, Chrome es conocido por su enfoque en la seguridad y actualizaciones automáticas, lo que garantiza que los usuarios siempre tengan acceso a las últimas mejoras y protecciones contra amenazas de seguridad (Google, 2023c). Su compatibilidad con una amplia gama de extensiones también permite personalizar y ampliar las funcionalidades del navegador, adaptándolo a las necesidades específicas del proyecto (Chrome Web Store, 2023).

En el contexto de la tesis, donde se busca desarrollar una aplicación web para el almacenamiento y análisis de datos en la industria petrolera, Google Chrome ofrece las herramientas necesarias para probar y desplegar la aplicación de manera efectiva. Su rendimiento superior, junto con sus capacidades de desarrollo y seguridad, lo convierten en la opción preferida para garantizar que la aplicación funcione de manera eficiente y segura.

### **3.3 Capítulo 3 – Desarrollo de la Aplicación**

En este capítulo, nos adentraremos en el proceso de desarrollo de la aplicación, un esfuerzo que requiere una planificación meticulosa y una ejecución precisa. Para lograr un sistema robusto y eficiente, se decidió dividir el desarrollo de la aplicación en tres componentes fundamentales: el cliente, el servidor y el controlador. Esta división no solo facilita la gestión del proyecto, sino que también asegura que cada parte del sistema pueda ser desarrollada y optimizada de manera independiente, permitiendo una integración más fluida y eficiente (Fowler, 2002).

El desarrollo del cliente se centra en la creación de una interfaz de usuario intuitiva y atractiva, que permita a los usuarios interactuar con la aplicación de manera sencilla y eficiente. Utilizando tecnologías modernas de desarrollo web, nos aseguramos de que la experiencia del usuario sea fluida y responsiva (Nielsen, 2000).

La configuración del servidor es crucial para garantizar que la aplicación pueda manejar múltiples solicitudes de manera simultánea, proporcionando datos de manera rápida y segura. Se estableció un gran énfasis en el establecimiento de una arquitectura de servidor robusta que pueda escalar según sea necesario, asegurando que los datos sean procesados y entregados de manera eficiente (Tanenbaum & Van Steen, 2007).

Finalmente, el desarrollo del controlador actúa como el puente entre el cliente y el servidor, gestionando la lógica de negocio y asegurando que las solicitudes del usuario se traduzcan en acciones concretas en el servidor. Este componente es esencial para mantener la coherencia y la integridad de los datos, asegurando que la aplicación funcione de acuerdo con los requisitos establecidos (Gamma et al., 1994).

A lo largo de este capítulo, exploraremos cada uno de estos componentes en detalle, destacando las decisiones de diseño y las tecnologías empleadas para construir una aplicación que no solo cumpla con las expectativas actuales, sino que también esté preparada para futuras expansiones y mejoras.

Para estructurar el código de manera efectiva durante el desarrollo de la aplicación, es esencial organizarlo en carpetas que separen las funciones del programa (Martin, 2008). Para esto se crearon dos carpetas principales: la carpeta `app` y la carpeta `resources`. La carpeta `app` contiene el código relacionado con la lógica de la aplicación y la interfaz del usuario, asegurando que todos los componentes de la aplicación estén bien organizados y fácilmente accesibles. Por otro lado, la carpeta `resources` se utiliza para almacenar recursos adicionales necesarios para el funcionamiento de la aplicación, como archivos de configuración, imágenes, y otros activos estáticos.

Además de estas carpetas principales, también se cuenta con archivos generales que se encuentran en la carpeta principal del proyecto. Estos archivos incluyen configuraciones globales y scripts que son esenciales para el funcionamiento general de la aplicación. Esta estructura de carpetas no solo facilita el mantenimiento del código, sino que también mejora la colaboración entre los desarrolladores al proporcionar un marco claro y organizado para el desarrollo del software (Hunt & Thomas, 1999). Esta organización es fundamental para asegurar que el proceso de desarrollo sea eficiente y que el código sea escalable y fácil de gestionar a medida que la aplicación crece y evoluciona.

### ***3.3.1 Desarrollo del Cliente***

En el desarrollo del cliente la aplicación, se estructuró el código de manera que se organiza en carpetas que reflejan las diferentes funcionalidades y características de la aplicación. Dentro de la carpeta principal del cliente, encontramos la carpeta `components`, que contiene todos los componentes de la interfaz de usuario. Estos componentes son la base de nuestra aplicación React, permitiendo una construcción modular y reutilizable de la interfaz.

Además de la carpeta `components`, se crearon carpetas específicas para cada una de las fuentes de energía que la aplicación maneja. Estas carpetas incluyen `Primary Energy`, `Carbon Dioxide`, `Oil`, `Gas`, `Coal`, `Nuclear Energy`, `Hydroelectricity`, `Renewables`, `Electricity Generation`, y `Key Materials`. Cada una de estas carpetas está diseñada para contener componentes y archivos relacionados con la visualización y gestión de datos específicos de cada tipo de energía.

#### **3.3.1.1 Components**

En esta carpeta encontramos los principales archivos visuales de la aplicación como el `Chart.tsx`, `SelectorBar.tsx`, `Sidebar.tsx`, `SidebarMenuItem.tsx`, `visor.tsx` y `wrapper.tsx`.

***Chart.tsx***

Este componente es responsable de renderizar los gráficos utilizados en la aplicación. Proporciona una representación visual de los datos, permitiendo a los usuarios interpretar tendencias y patrones de manera intuitiva.

***SelectorBar.tsx***

Actúa como una barra de selección que permite a los usuarios filtrar y seleccionar diferentes conjuntos de datos o categorías dentro de la aplicación. Este componente mejora la interactividad y la personalización de la experiencia del usuario.

***Sidebar.tsx y SidebarMenuItem.tsx***

Estos componentes trabajan juntos para crear un menú lateral que facilita la navegación dentro de la aplicación. La barra lateral proporciona acceso rápido a diferentes secciones, mientras que los elementos del menú permiten una navegación detallada y específica.

***visor.tsx***

Este componente se encarga de mostrar los datos seleccionados en un formato claro y accesible. Funciona como una ventana de visualización donde los usuarios pueden explorar la información en detalle.

***Wrapper.tsx***

Actúa como un contenedor general para otros componentes, asegurando que se mantenga una estructura coherente y estilizada a lo largo de toda la aplicación. Este componente es esencial para mantener la consistencia del diseño y facilitar la integración de nuevos elementos.

### 3.3.1.2 Carpetas Fuentes de Energía

En la sección Components del Capítulo 3, además de los archivos visuales principales mencionados anteriormente, es importante destacar la estructura organizada para las diferentes fuentes de energía. Dentro de la carpeta principal, encontramos subcarpetas dedicadas a cada tipo de energía, como primary para "Primary Energy", carbon para "Carbon Dioxide", oil para "Petróleo", gas, coal para "Carbon", nuclear para "Nuclear Energy", Hydroelectricity para "Hidroelectricidad", renewable para "Renovables", electric-generation para "Generación de energía", key-materials para "Materiales Claves" y multi-graph para el apartado multi grafica.

Cada una de estas carpetas de energía contiene un único archivo llamado page.tsx. Este enfoque de estructura uniforme facilita la consistencia y mantenibilidad del código. El archivo page.tsx en cada carpeta sirve como punto de entrada para la visualización y funcionalidad específica de esa fuente de energía, a continuación, mostramos el código del archivo page.tsx antes mencionado para las emisiones de dióxido de carbono:

```
import { Metadata } from "next";
import SelectorBar from "../components/SelectorBar";
import { Carbon } from "../utils/optionsRealData";
import Chart from "@app/components/Chart";
import Wrapper from '../components/Wrapper';

export const metadata: Metadata = {
  title: "Carbon Dioxide",
  description: "Page with data of carbon dioxide usage",
};

export default function CarbonPage() {
  return (
    <div className="w-full min-h-screen">
      <Wrapper datos={Carbon}></Wrapper>
    </div>
  );
}
```

La estructura del archivo `page.tsx` es similar en todas las carpetas de energía, lo que permite una fácil navegación y comprensión del código. Cada `page.tsx` importa y utiliza los componentes visuales necesarios para representar los datos y gráficos relacionados con su fuente de energía específica. Esta organización modular permite una clara separación de preocupaciones, facilitando el desarrollo y mantenimiento de cada sección de energía de manera independiente.

Esta arquitectura no solo mejora la organización del código, sino que también permite una fácil expansión del sistema en el futuro. Si se necesita agregar nuevas fuentes de energía o modificar las existentes, la estructura establecida proporciona un marco claro para integrar estos cambios de manera coherente con el resto de la aplicación.

La importación de todos los objetos visuales en cada `page.tsx` asegura que cada sección de energía tenga acceso a los componentes necesarios para una representación completa y detallada de sus datos específicos, manteniendo al mismo tiempo una estructura de código limpia y organizada.

### ***3.3.2 Configuración del Servidor***

En la configuración del servidor de la aplicación, se optó por utilizar MongoDB como sistema de gestión de base de datos. Esta elección proporciona una flexibilidad y escalabilidad excepcionales, perfectas para manejar los grandes volúmenes de datos complejos y no estructurados típicos de la industria petrolera.

Dentro de la estructura del servidor, encontramos la carpeta “resources”, que juega un papel crucial en la organización y almacenamiento de nuestros datos. Esta carpeta contiene dos elementos principales:

- **Base de datos BP Statistical Review 2022:** Esta base de datos se compone de múltiples archivos CSV, cada uno correspondiente a una hoja específica del informe original de BP. La decisión de mantener cada hoja como un archivo CSV separado nos permite una gestión más granular de los datos y facilita las actualizaciones parciales cuando sea necesario. Estos archivos CSV contienen información detallada sobre diversos aspectos de la industria energética global, incluyendo producción, consumo, reservas y tendencias de diferentes fuentes de energía.
- **Archivo pbData.power.json:** Este archivo JSON es una pieza clave en nuestra configuración. Contiene metadatos y estructuras de datos que ayudan a organizar y dar contexto a la información almacenada en los archivos CSV. Puede incluir información como esquemas de datos, relaciones entre diferentes conjuntos de datos, y configuraciones específicas para el procesamiento y visualización de los datos en nuestra aplicación web.

La combinación de estos recursos en la carpeta “resources” nos permite mantener una estructura de datos robusta y flexible. MongoDB, con su capacidad para manejar documentos JSON de forma nativa, nos facilita la integración de estos datos en la aplicación, permitiendo realizar consultas complejas y análisis detallados de manera eficiente.

Esta configuración del servidor no solo nos proporciona una base sólida para el almacenamiento y gestión de datos, sino que también posiciona a la aplicación favorablemente para futuras expansiones y actualizaciones de la base de datos, manteniendo la integridad y accesibilidad de la información crucial para la aplicación en el sector energético.

### 3.3.3 *Desarrollo del Controlador*

El desarrollo del controlador en esta aplicación es una parte crucial que se encarga de manejar la lógica de negocio y la comunicación entre el cliente y la base de datos. La estructura del controlador se organiza de la siguiente manera:

Dentro de la carpeta "app", encontramos la subcarpeta "api", que contiene la lógica principal para el manejo de las solicitudes del cliente. Esta carpeta "api" se divide en dos subcarpetas principales:

- **region:** Esta carpeta contiene un archivo "route.ts" que se encarga de definir las rutas y los controladores específicos para manejar las solicitudes relacionadas con las regiones. Aquí se implementan las funciones que recuperan y procesan los datos de regiones desde la base de datos.
- **stats:** Similar a la carpeta "region", esta contiene un archivo "route.ts" que define las rutas y controladores para manejar las solicitudes relacionadas con las estadísticas. Este archivo implementa la lógica para recuperar y procesar los datos estadísticos de la base de datos.

Además de la carpeta "api", el controlador también incluye una carpeta "utils" que contiene tres archivos importantes:

- **mongodb.js:** Este archivo contiene la configuración y las funciones necesarias para establecer y manejar la conexión con la base de datos MongoDB. Aquí se definen las operaciones básicas de lectura y escritura en la base de datos.
- **optionsData.js:** Este archivo contiene datos estructurados que definen las opciones de regiones y países para la aplicación.

- **optionsRealData.js:** Este archivo contiene definiciones de opciones para diferentes tipos de datos energéticos. Específicamente:
  1. Define constantes para varias categorías de datos energéticos como primaryEnergy, Carbon, Oil, Gas, Coal, Nuclear, HydroElectricity, ElectricityGeneration, KeyMaterial, y RenewableEnergy.
  2. Cada categoría es un array de objetos, donde cada objeto representa una opción específica de datos con propiedades como value (valor interno), label (etiqueta descriptiva), y query (consulta asociada).
  3. Estas opciones cubren una amplia gama de métricas energéticas, incluyendo consumo, producción, emisiones, y generación de electricidad, entre otras.
  4. Las unidades de medida están claramente especificadas en las etiquetas, como Exajoules [EJ], Million Tonnes [Mte], Terawatt hour [Twh], etc.

La estructura del controlador está diseñada para separar claramente las responsabilidades: las rutas y la lógica de negocio se manejan en los archivos "route.ts" dentro de las carpetas "region" y "stats", mientras que las utilidades y configuraciones generales se mantienen en la carpeta "utils". Esta organización permite una mejor mantenibilidad y escalabilidad del código, facilitando la adición de nuevas funcionalidades o la modificación de las existentes sin afectar otras partes del sistema.

### **3.3.4 Pruebas Beta**

Durante la fase de pruebas beta de la aplicación, se llevó a cabo un riguroso proceso de validación para asegurar la precisión y fiabilidad de los datos presentados. Un componente clave

de estas pruebas fue la comparación de gráficas generadas por la aplicación con gráficas creadas manualmente a partir de los datos originales de la base de datos BP Statistical Review 2022.

Para este proceso, se seleccionó aleatoriamente varias hojas de la base de datos, abarcando diferentes aspectos de la industria energética, como producción de petróleo, consumo de gas natural, emisiones de CO<sub>2</sub> y generación de energía renovable. A partir de estos datos, se crearon gráficas de referencia utilizando herramientas estándar de visualización de datos.

Paralelamente, se utilizó la aplicación para generar gráficas basadas en los mismos conjuntos de datos. El objetivo era comparar estas gráficas generadas automáticamente con las gráficas de referencia, evaluando aspectos como la precisión de los valores representados, la correcta visualización de tendencias temporales y la adecuada representación de diferentes regiones o países.

Este proceso de comparación permitió no solo verificar la exactitud de los datos presentados por la aplicación, sino también evaluar la claridad y efectividad de las visualizaciones. Además, ayudó a identificar y corregir cualquier discrepancia o error en el procesamiento de datos, asegurando así la integridad de la información presentada a los usuarios.

Las pruebas beta también incluyeron una evaluación de la usabilidad de la interfaz y la capacidad de la aplicación para manejar diferentes tipos de consultas y filtros. Esto permitió refinar la experiencia del usuario y optimizar el rendimiento de la aplicación bajo diversas condiciones de uso.

Los resultados de estas pruebas fueron fundamentales para validar la fiabilidad de la aplicación y proporcionó valiosos insights para futuras mejoras y desarrollos. En las siguientes

secciones, se presentan algunos ejemplos específicos de estas comparaciones, demostrando cómo la aplicación logra reproducir con precisión los datos de la BP Statistical Review 2022, al tiempo que ofrece una interfaz intuitiva y herramientas de análisis avanzadas.

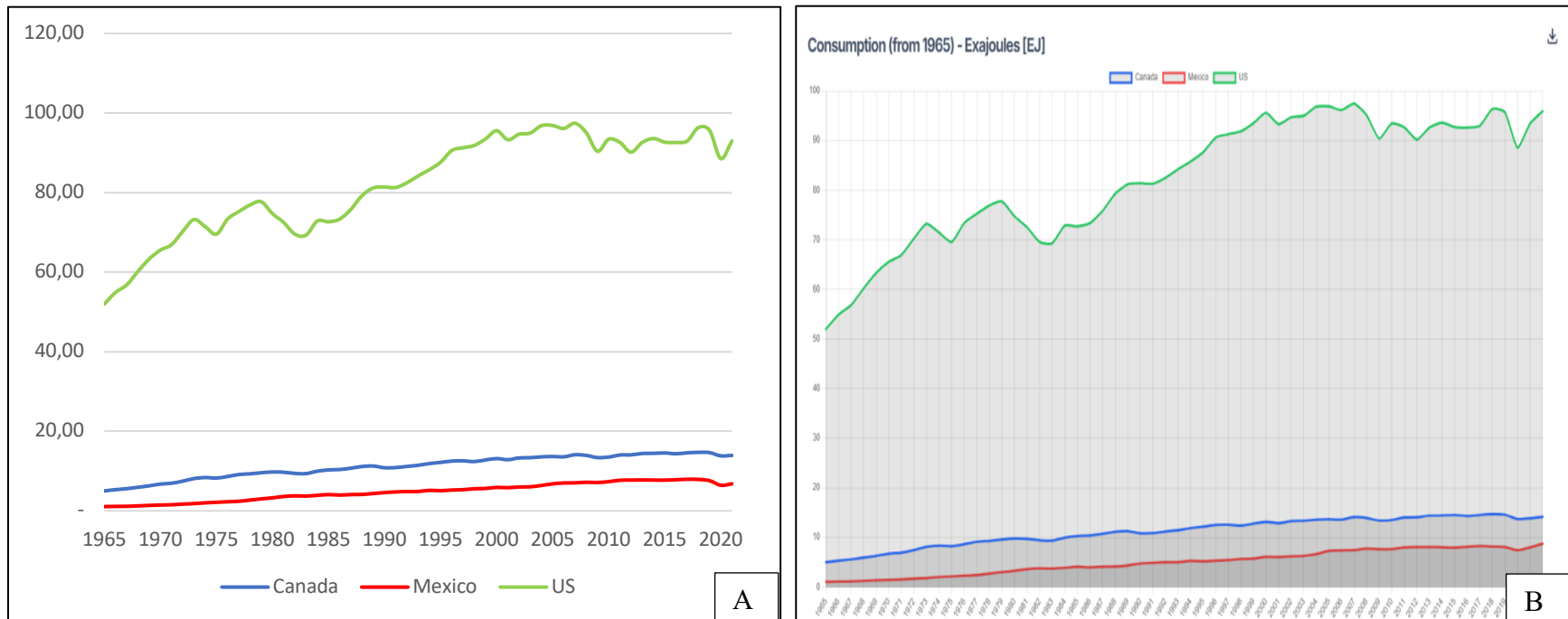
#### **3.3.4.1 Comparación de graficas**

Para determinar un número aceptable de hojas a revisar en una prueba beta de una aplicación que compara gráficas creadas en Excel con las generadas por la aplicación, es importante considerar varios factores:

- **Tamaño de la Base de Datos:** Si la base de datos es extensa, revisar un pequeño porcentaje de las hojas podría ser suficiente para obtener una representación adecuada de la precisión y coherencia de las gráficas (Patton, 2006).
- **Diversidad de Datos:** Asegúrate de que las hojas seleccionadas cubran una variedad de tipos de datos y categorías relevantes para la aplicación. Esto ayudará a identificar problemas específicos en diferentes áreas del sistema (Myers et al., 2011).
- **Recursos Disponibles:** Considera el tiempo y los recursos disponibles para realizar las pruebas. Un muestreo más pequeño puede ser más manejable si los recursos son limitados (Sommerville, 2016).
- **Prácticas Comunes:** En la industria, un muestreo del 10% al 20% de las hojas puede ser considerado adecuado para pruebas beta, siempre que las hojas seleccionadas sean representativas de todo el conjunto de datos (Leffingwell & Widrig, 2003).

*Primary Energy: Consumption – Exajoules (from 1965) – North America*

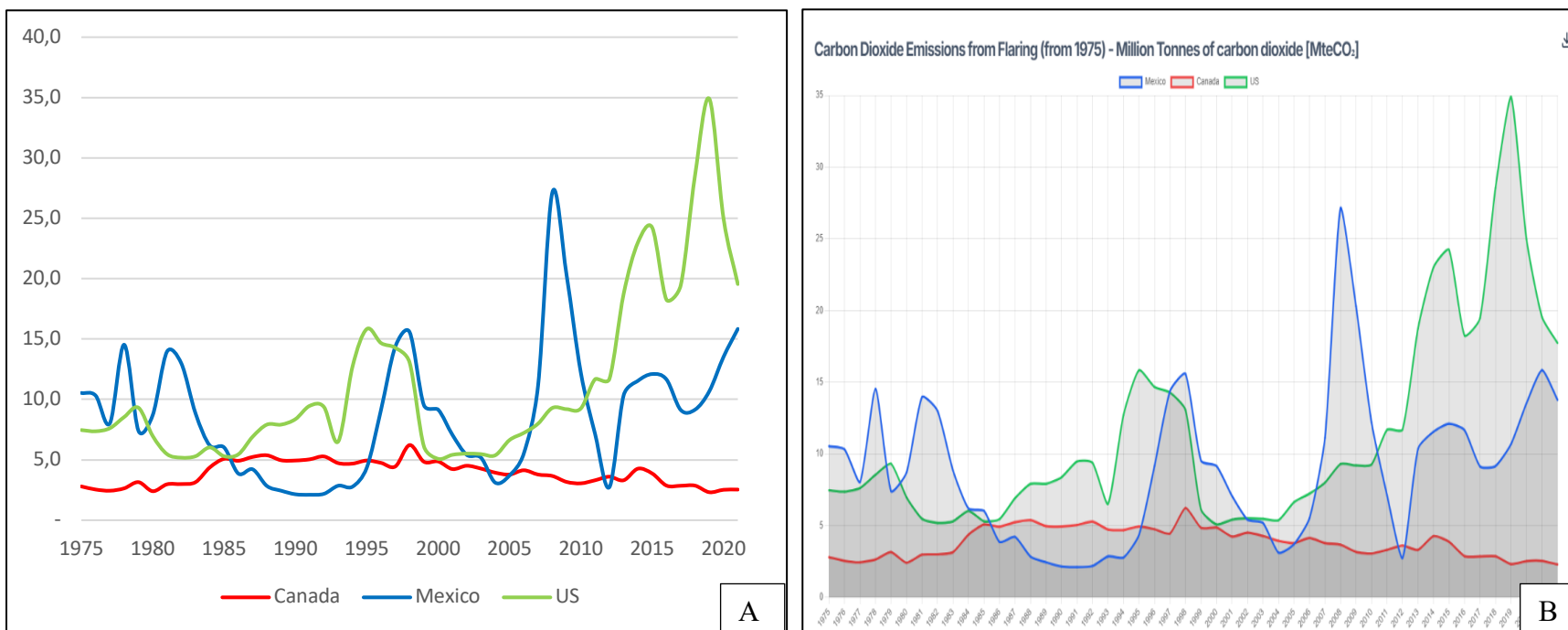
**Figura 9** – Comparativo consumo energías primarias en Exajulios desde 1965 para Norte América



Nota: Estas imágenes muestran un comparativo entre las gráficas generadas en Excel (A) y las gráficas generadas por el programa EnergyMetrics (B). Este comparativo permite visualizar la data desde 1965 para el consumo de energías primarias en la región de Norte América.

*Carbon Dioxide Emissions from Flaring – Million tonnes of carbon dioxide (from 1975) – North America*

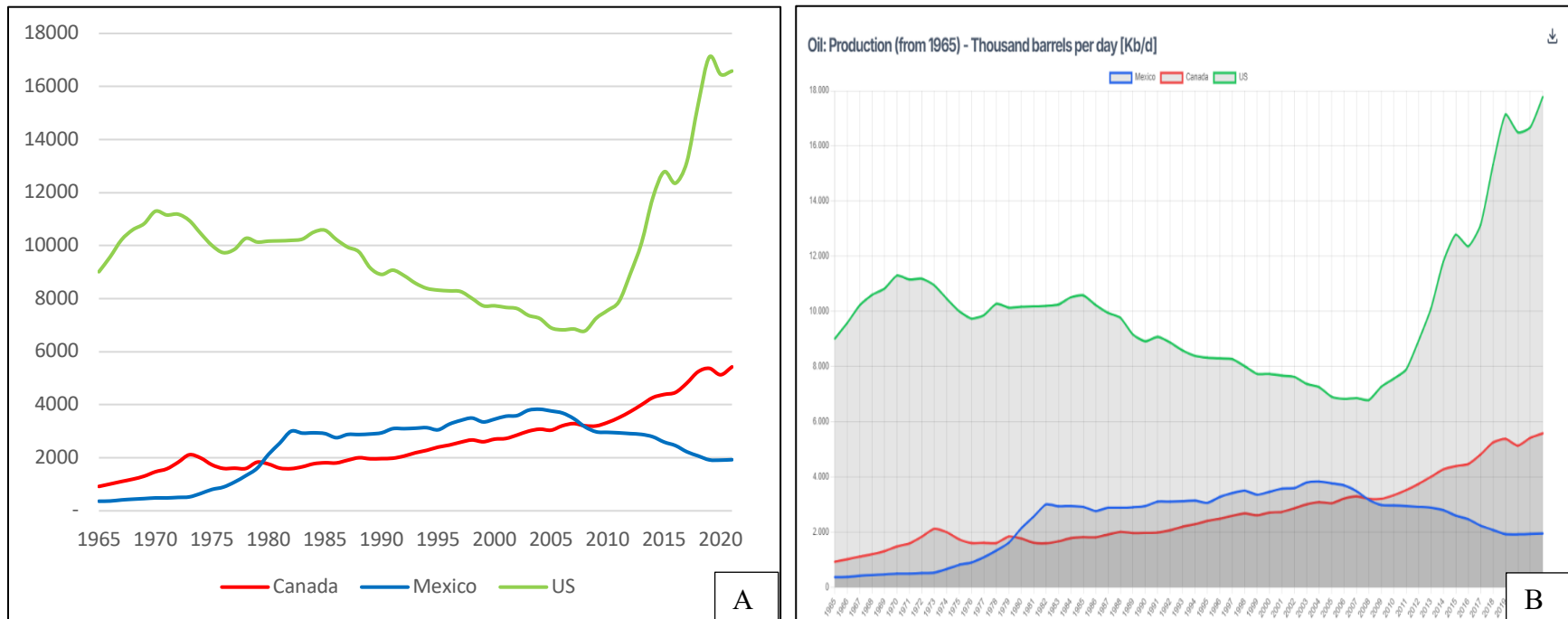
**Figura 10** – *Comparativo emisiones de dióxido de carbono provenientes de la quema en antorcha en millones de toneladas desde 1975 para Norte América.*



Nota: Estas imágenes muestran un comparativo entre las gráficas generadas en Excel (A) y las gráficas generadas por el programa EnergyMetrics (B). Este comparativo permite visualizar la data desde 1975 para las emisiones de dióxido de carbono provenientes de la quema en antorcha en la región de Norte América

**Oil: Production - Barrels (from 1965) – North America**

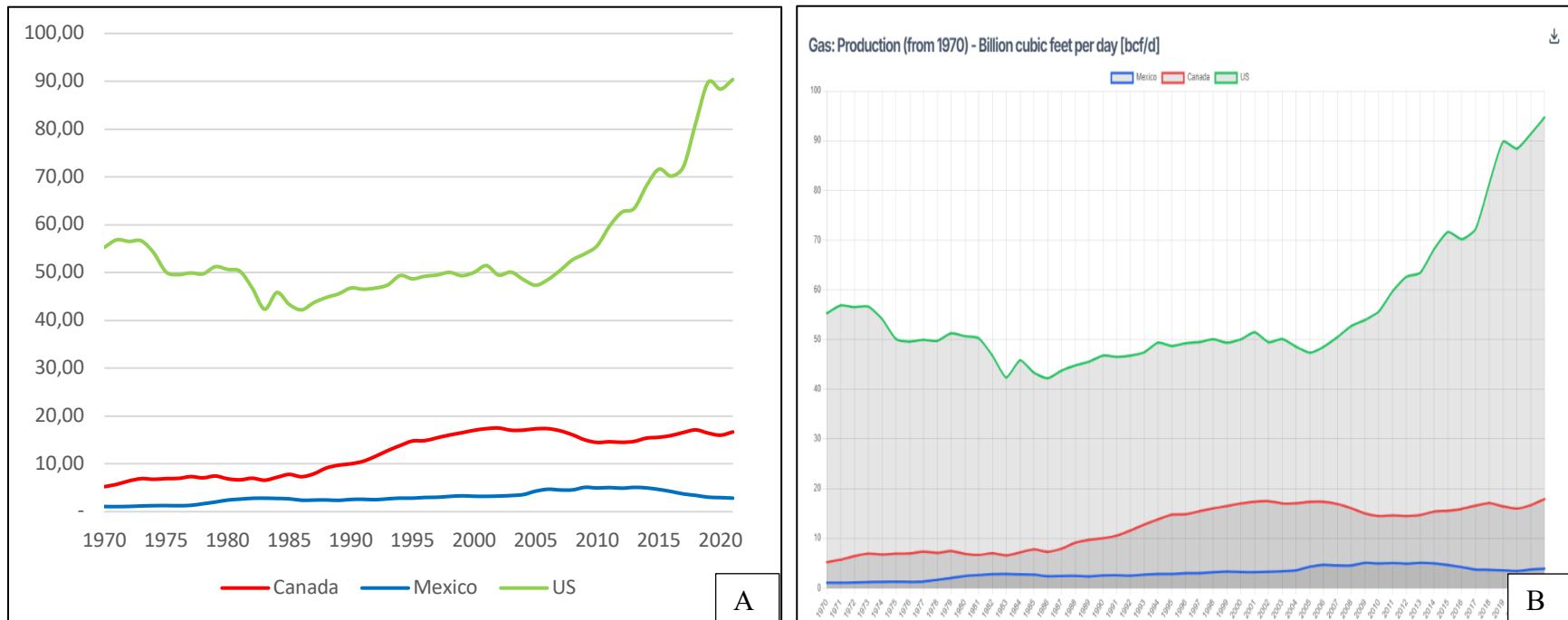
**Figura 11 – Comparativo producción de petróleo en miles de barriles desde 1965 para Norte América**



Nota: Estas imágenes muestran un comparativo entre las gráficas generadas en Excel (A) y las gráficas generadas por el programa EnergyMetrics (B). Este comparativo permite visualizar la data desde 1965 para la producción de petróleo en la región de Norte América

**Gas: Production – Bcf (from 1970) – North America**

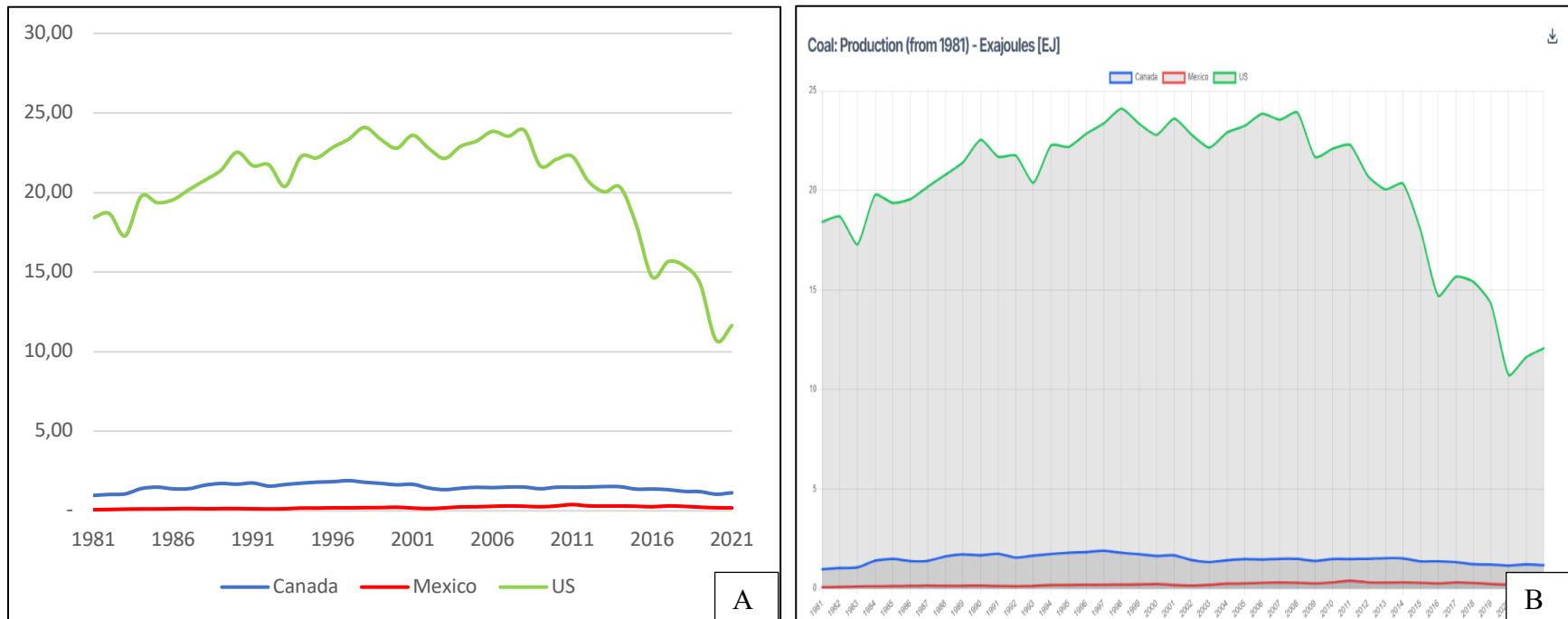
**Figura 12** – Comparativo producción de gas en miles de millones de pies cúbicos desde 1970 para Norte América.



Nota: Estas imágenes muestran un comparativo entre las gráficas generadas en Excel (A) y las gráficas generadas por el programa EnergyMetrics (B). Este comparativo permite visualizar la data desde 1975 para la producción de gas en la región de Norte América

*Coal: Production – Exajoules (from 1981) – North America*

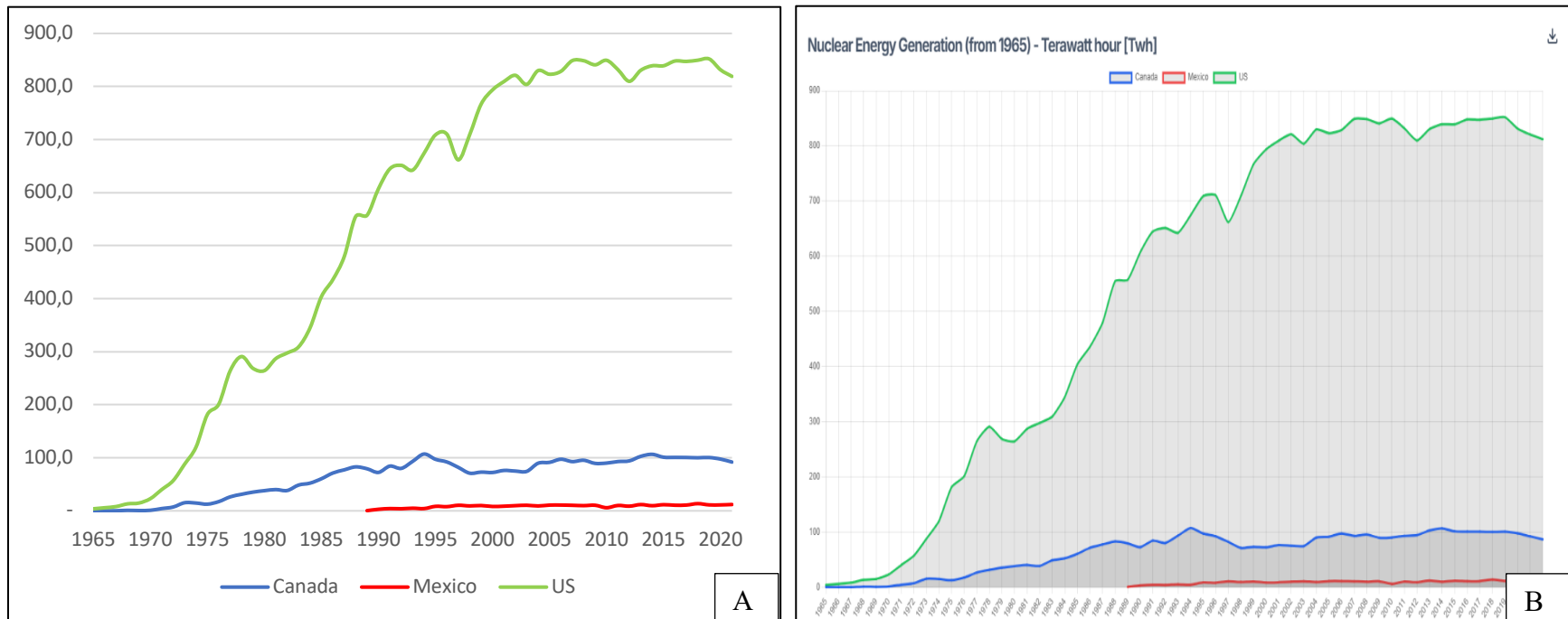
**Figura 13** – Comparativo producción de carbón en Exajulios desde 1981 para Norte América



Nota: Estas imágenes muestran un comparativo entre las gráficas generadas en Excel (A) y las gráficas generadas por el programa EnergyMetrics (B). Este comparativo permite visualizar la data desde 1981 para la producción de carbón en la región de Norte América

*Nuclear Energy - Generation – TWh (from 1965) – North America*

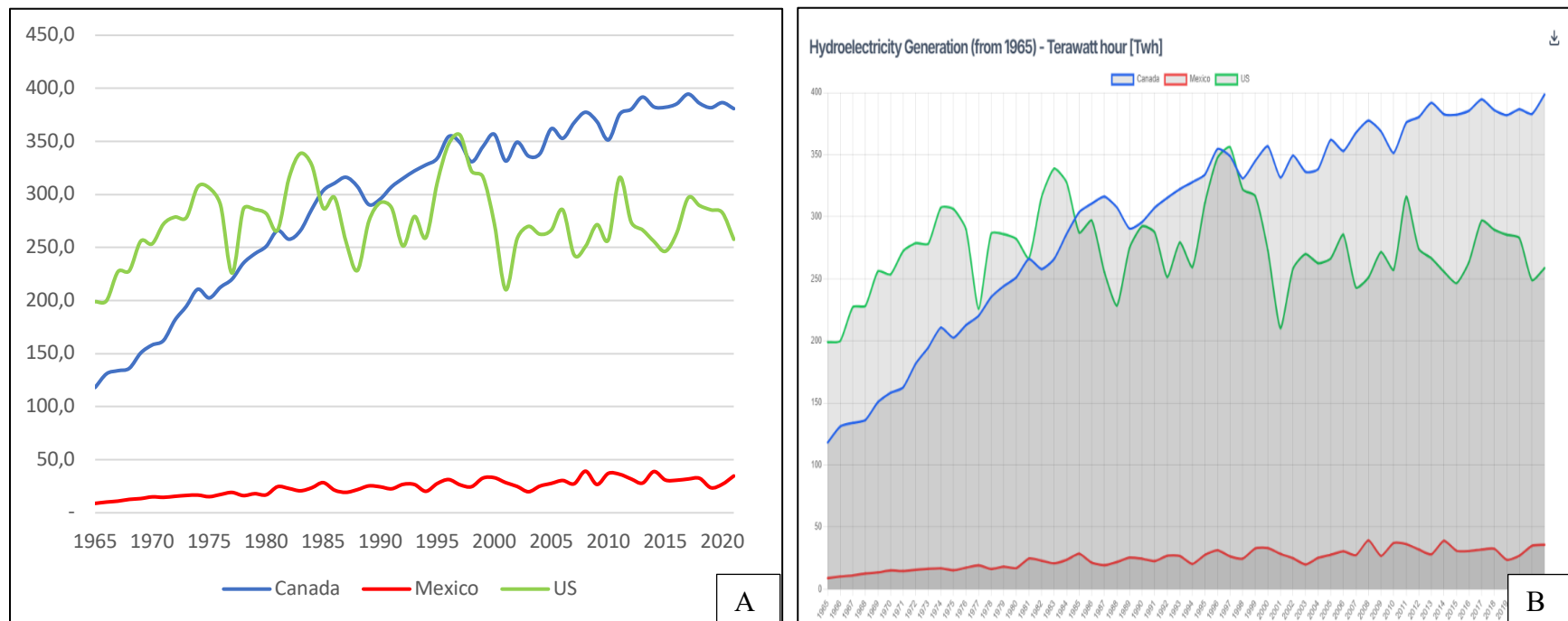
**Figura 14** – Comparativo generación de energía nuclear en Teravatio-hora desde 1965 para Norte América.



Nota: Estas imágenes muestran un comparativo entre las gráficas generadas en Excel (A) y las gráficas generadas por el programa EnergyMetrics (B). Este comparativo permite visualizar la data desde 1965 para generación de energía nuclear en la región de Norte América

*Hydroelectricity - Generation – TWh (from 1965) – North America*

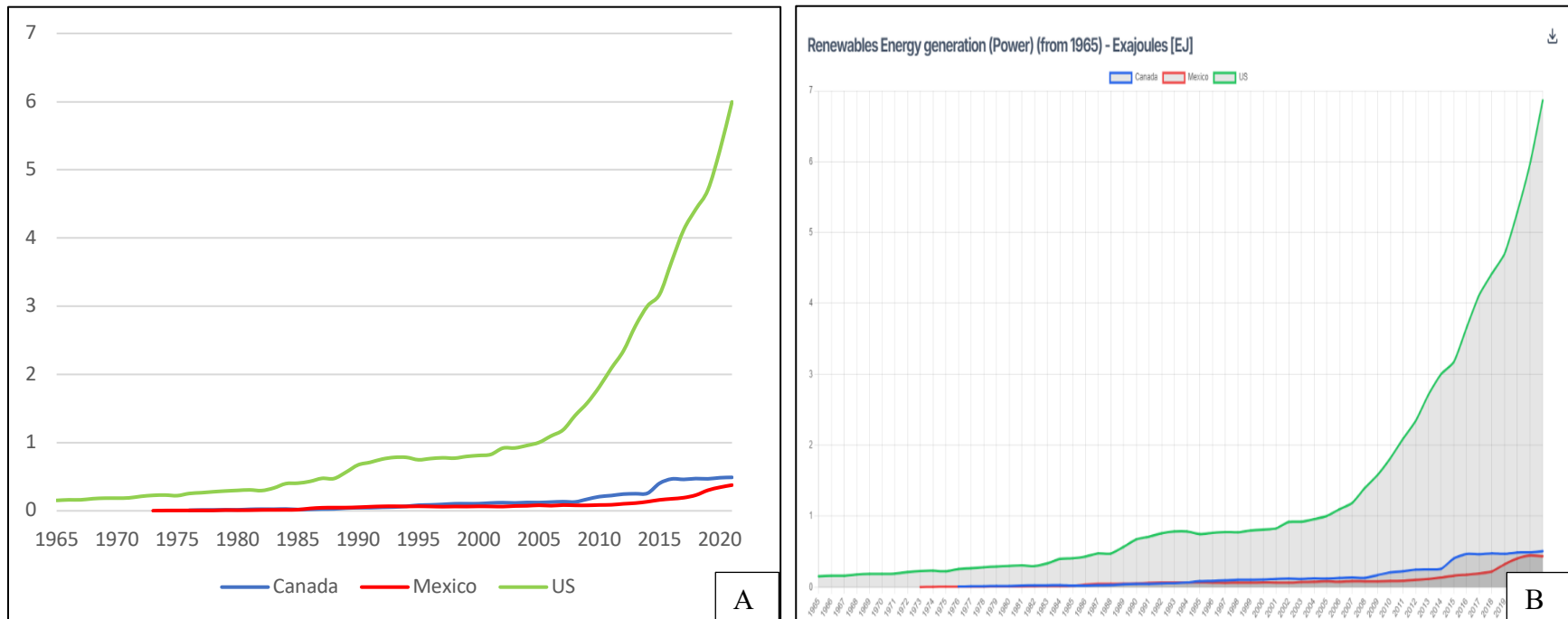
**Figura 15** – Comparativo generación de hidroelectricidad en Teravatio-hora desde 1965 para Norte América.



Nota: Estas imágenes muestran un comparativo entre las gráficas generadas en Excel (A) y las gráficas generadas por el programa EnergyMetrics (B). Este comparativo permite visualizar la data desde 1965 para generación de hidroelectricidad en la región de Norte América

*Renewables – Renewable Power – Exajoules (from 1965) – North America*

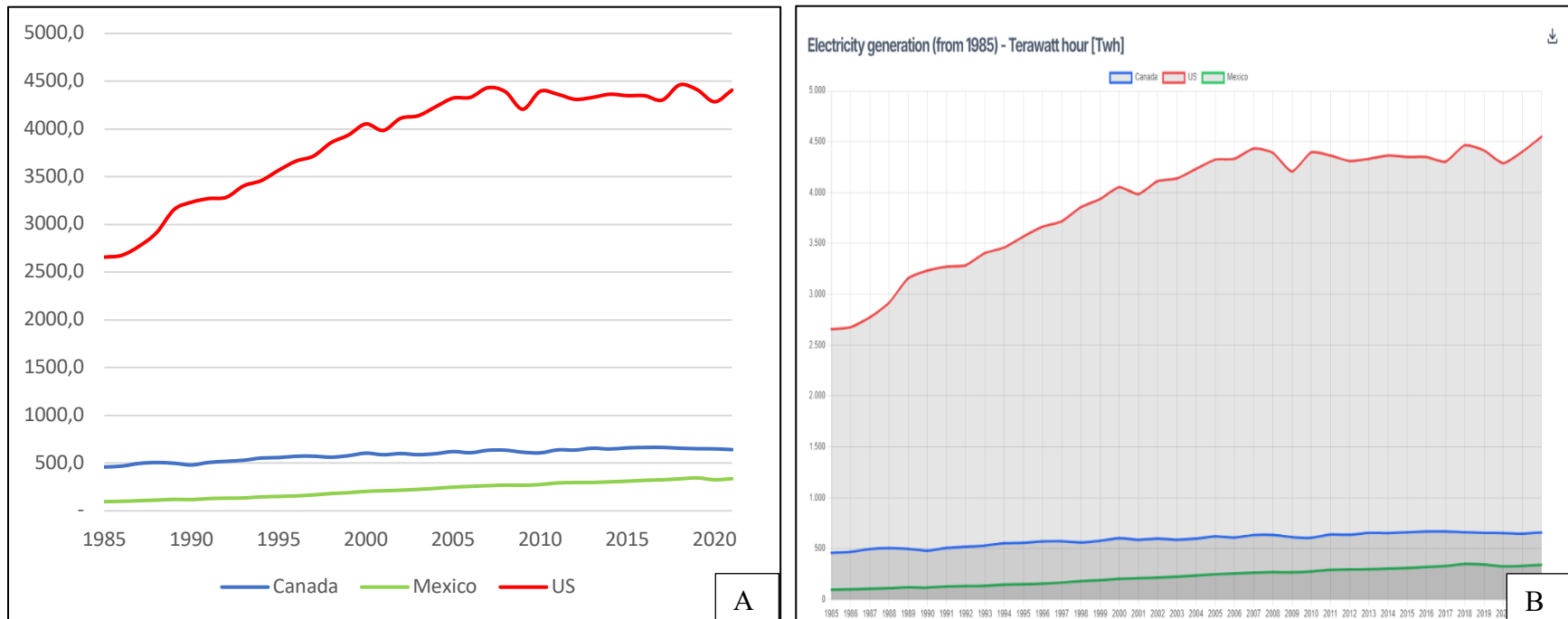
**Figura 16** – Comparativo generación de energía renovable en Exajulios desde 1965 para Norte America.



Nota: Estas imágenes muestran un comparativo entre las gráficas generadas en Excel (A) y las gráficas generadas por el programa EnergyMetrics (B). Este comparativo permite visualizar la data desde 1965 para generación de energía renovable en la región de Norte América

*Electricity generation - TWh (from 1985) – North America*

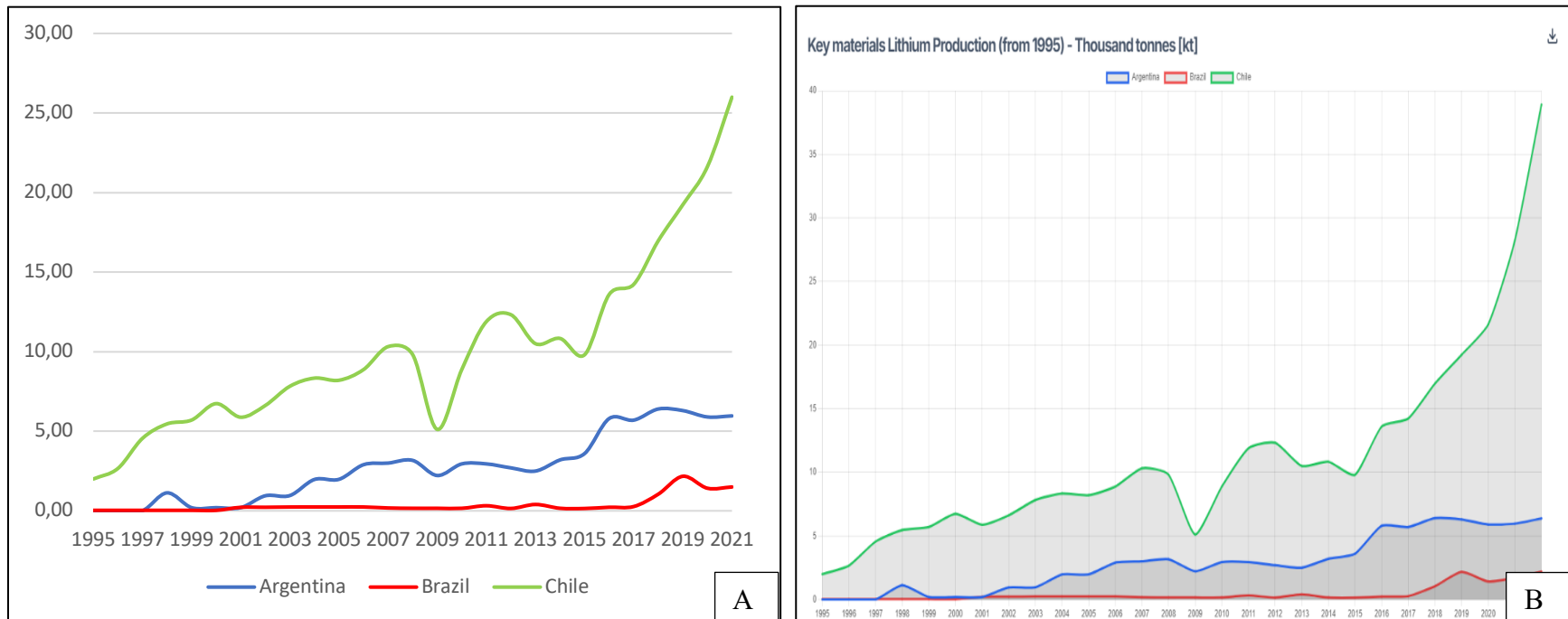
**Figura 17** – Comparativo generación de electricidad en Teravatio-hora desde 1985 para Norte América



Nota: Estas imágenes muestran un comparativo entre las gráficas generadas en Excel (A) y las gráficas generadas por el programa EnergyMetrics (B). Este comparativo permite visualizar la data desde 1965 para generación de electricidad en la región de Norte América

*Key materials - Lithium Production Thousand tonnes (from 1995) – South & Central America*

**Figura 18** – Comparativo producción de Litio en miles de toneladas desde 1995 para Norte América.



Nota: Estas imágenes muestran un comparativo entre las gráficas generadas en Excel (A) y las gráficas generadas por el programa EnergyMetrics (B). Este comparativo permite visualizar la data desde 1995 para la producción de litio en la región de Sur y Centro América.

### 3.3.4.2 Resultados de la prueba beta

Durante las pruebas beta de la aplicación, se realizó un análisis exhaustivo para evaluar su capacidad de reproducir gráficas de manera precisa y coherente. Fueron seleccionadas aleatoriamente 10 hojas de la base de datos BP Statistical Review 2022, cubriendo una variedad de categorías como Energía Primaria, Emisiones de Dióxido de Carbono, Petróleo, Gas, Carbón, Energía Nuclear, Hidroelectricidad, Energías Renovables, Generación de Electricidad y Materiales Clave. Esta selección estratégica permitió evaluar la aplicación en un amplio espectro de datos energéticos.

Los resultados de las pruebas fueron altamente satisfactorios. La aplicación logró reproducir las gráficas con un 100% de efectividad, coincidiendo perfectamente con las gráficas generadas manualmente a partir de los datos originales en Excel. Este éxito se debe a la robustez del sistema de gestión de datos implementado, que asegura la integridad y precisión de la información procesada.

La capacidad de la aplicación para manejar datos complejos y generar visualizaciones precisas es un testimonio de su diseño eficiente y su potencial para ser una herramienta valiosa en el análisis de datos energéticos. Estos resultados no solo validan la funcionalidad de la aplicación, sino que también refuerzan su utilidad en contextos profesionales y educativos, donde la precisión y la coherencia de los datos son cruciales.

En conclusión, las pruebas beta han demostrado que la aplicación está bien equipada para cumplir con los requisitos del sector energético, proporcionando una plataforma confiable para el análisis y la visualización de datos críticos. Estos resultados positivos nos motivan a continuar mejorando y expandiendo las capacidades de la aplicación en futuras iteraciones.

### **3.4 Capítulo 4 – Resultados, Beneficios y Manual de Usuario**

#### **3.4.1 Resultados**

La aplicación se ha construido con un enfoque centralizado, utilizando la base de datos BP Statistical Review 2022 como su principal fuente de datos. Esta centralización ha permitido un manejo eficiente de datos históricos y contextuales del sector energético, asegurando que la información utilizada sea coherente y actualizada. La elección de la base de datos BP ha sido estratégica, dado su reconocimiento y fiabilidad en el sector energético, proporcionando una base sólida para el análisis de tendencias y patrones históricos.

Gracias a la implementación de MongoDB como sistema de gestión de bases de datos, la aplicación ha demostrado una notable capacidad para manejar grandes volúmenes de datos complejos y no estructurados. MongoDB ha ofrecido flexibilidad y escalabilidad, permitiendo que la aplicación se adapte eficientemente a las necesidades cambiantes del sector petrolero. Su capacidad para almacenar y recuperar datos de manera rápida y confiable ha sido fundamental para respaldar procesos críticos dentro de la aplicación.

Es importante señalar que, en su estado actual, la aplicación no procesa datos en tiempo real. Sin embargo, se ha establecido una base sólida para futuras mejoras en esta área. En trabajos futuros, se planea integrar un módulo que permita a las empresas del sector monitorear datos en tiempo real, lo que proporcionará una herramienta valiosa para la toma de decisiones estratégicas y la optimización de operaciones.

En conclusión, los resultados obtenidos hasta ahora demuestran la eficacia de la aplicación en el manejo de datos complejos y su capacidad para proporcionar una infraestructura robusta y eficiente para el análisis de datos en la industria petrolera y educativa. Estos resultados subrayan

la importancia de integrar herramientas tecnológicas avanzadas en el desarrollo de soluciones innovadoras para sectores tradicionales.

### 3.4.2 *Beneficios*

Para abordar los beneficios del aplicativo web es necesario realizar un comparativo entre el aplicativo web desarrollado versus otros recursos digitales, además de lo anterior es importante considerar varios aspectos que hacen que esta aplicación sea única y valiosa, especialmente en el contexto educativo y del sector petrolero.

#### **Beneficios del Aplicativo:**

1. ***Centralización de Datos Confiables:*** La aplicación está centralizada en la base de datos BP Statistical Review 2022, lo que proporciona un acceso a datos históricos y contextuales altamente confiables y reconocidos en el sector energético. Esto asegura que los usuarios trabajen con información precisa y actualizada, algo que no siempre es garantizado por otros recursos digitales que pueden depender de múltiples fuentes de datos menos consistentes.
2. ***Plataforma Educativa Integral:*** Para estudiantes y docentes, la aplicación ofrece una herramienta robusta para explorar y analizar datos reales del sector energético. Esto facilita una comprensión más profunda de la evolución de las energías y prepara a los futuros ingenieros de petróleo para enfrentar desafíos reales. Otros recursos digitales pueden no ofrecer la misma profundidad de datos históricos ni la capacidad de personalizar el análisis según las necesidades educativas específicas.
3. ***Flexibilidad y Escalabilidad:*** Gracias a la implementación de MongoDB, la aplicación maneja eficientemente grandes volúmenes de datos no estructurados, adaptándose a las

necesidades cambiantes del sector. Esto la diferencia de otros recursos que pueden estar limitados por arquitecturas de bases de datos menos flexibles.

4. ***Preparación para Mejoras Futuras:*** La aplicación está diseñada con una arquitectura que permite futuras expansiones, como la integración de módulos para monitoreo de datos en tiempo real y el uso de inteligencia artificial para análisis avanzados. Esto asegura que la aplicación pueda evolucionar con las necesidades del sector, a diferencia de otros recursos que podrían requerir rediseños significativos para incorporar nuevas tecnologías.
5. ***Enfoque en la Toma de Decisiones Estratégicas:*** La futura capacidad de monitorear datos en tiempo real proporcionará a las empresas del sector petrolero una herramienta poderosa para la optimización de operaciones y la toma de decisiones estratégicas. Esta capacidad proyectada coloca al aplicativo en una posición ventajosa frente a otros recursos que no ofrecen esta funcionalidad.

#### **Comparación con Otros Recursos Digitales:**

1. ***Recursos Genéricos vs. Especializados:*** Mientras que muchos recursos digitales ofrecen datos generales sobre la industria energética, el aplicativo está específicamente diseñado para integrar datos relevantes y específicos del sector petrolero, lo que lo hace más útil para usuarios en este campo.
2. ***Interactividad y Personalización:*** La capacidad del aplicativo para personalizar la experiencia del usuario, especialmente en un contexto educativo, lo diferencia de otros recursos que pueden ofrecer experiencias más estáticas y menos adaptadas a las necesidades individuales de los usuarios.

3. **Capacidades Analíticas Avanzadas:** Con la integración futura de inteligencia artificial, el aplicativo ofrecerá capacidades analíticas que superan a las de muchos otros recursos digitales, permitiendo insights más profundos y personalizados.

En el ámbito educativo, los beneficios serán significativos. La aplicación proporcionará a estudiantes y docentes una plataforma robusta para explorar y analizar datos reales del sector energético. La integración de la base de datos BP Statistical Review 2022 permitirá a los estudiantes de ingeniería petrolera acceder a datos históricos, facilitando una comprensión más profunda de la evolución de las energías a lo largo del tiempo. Esto enriquecerá el proceso educativo, permitiendo a los estudiantes aplicar teorías aprendidas en el aula a situaciones del mundo real.

Para los docentes, la aplicación servirá como una herramienta didáctica invaluable, permitiendo la incorporación de datos reales en el currículo académico y fomentando un enfoque más analítico y basado en datos en la enseñanza. La capacidad de la aplicación para integrar y analizar datos de diversas fuentes en trabajos futuros proporcionará una visión integral del panorama energético global, lo cual es crucial para la formación de futuros ingenieros de petróleo.

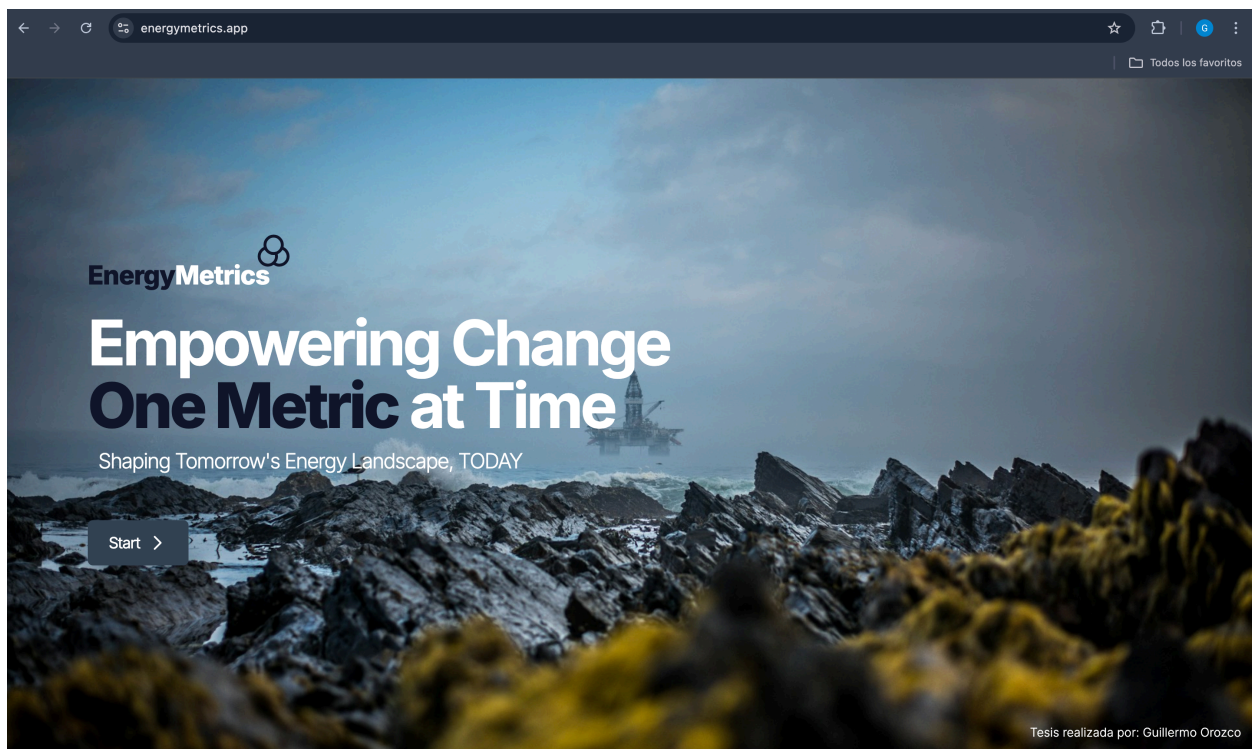
### 3.4.3 *Manual de Usuario*

En este manual de usuario, se ha tomado la decisión deliberada de omitir los títulos en las imágenes y mantener una numeración completamente aislada al resto del documento teniendo en cuenta que este se utilizara como documento de apoyo para el aplicativo. Esta elección se basa en consideraciones prácticas y pedagógicas, con el objetivo de facilitar una comprensión más fluida y directa del contenido visual. La inclusión de títulos en cada imagen podría potencialmente interrumpir el flujo de lectura y complicar innecesariamente la interpretación de las

representaciones visuales. En su lugar, se ha optado por integrar las descripciones y explicaciones pertinentes directamente en el texto que acompaña a cada imagen, permitiendo así una asociación más inmediata y contextualizada entre el contenido visual y su significado. Esta aproximación busca optimizar la experiencia del usuario, promoviendo una asimilación más eficiente y efectiva de la información presentada.

### 1. Como Ingresar al Programa

Para acceder al programa, primero abra su navegador web de preferencia. Para obtener las mejores funcionalidades, se recomienda encarecidamente utilizar Google Chrome, ya que el programa fue desarrollado específicamente para este navegador. Luego, en la barra de direcciones, ingrese la URL `www.energymetrics.app` y presione la tecla "Enter" o haga clic en el botón de búsqueda para cargar la página. Una vez que se haya cargado completamente, estará en la página de inicio del programa como se muestra en la siguiente figura.

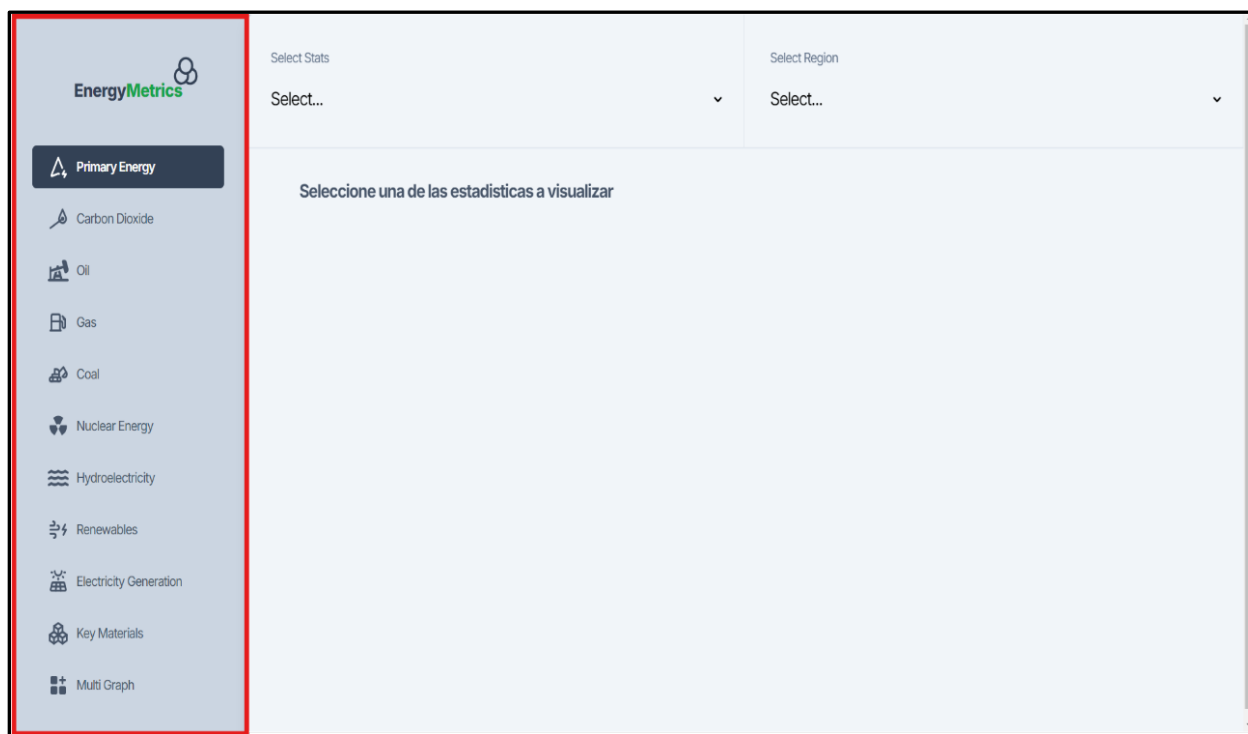


Para comenzar a utilizar la aplicación, simplemente localice y haga clic en el botón "Start" que aparece en la pantalla. Siguiendo estos pasos, podrá acceder y comenzar a utilizar el programa de manera rápida y sencilla.

## 2. Partes del Programa

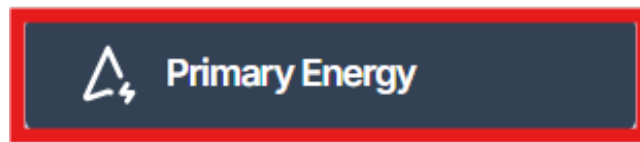
### 2.1 Side Nav

El Side Nav es el navegador lateral que se encuentra en lateral izquierdo del programa, la opción que seleccione en el navegador se tornara de un color más oscuro.



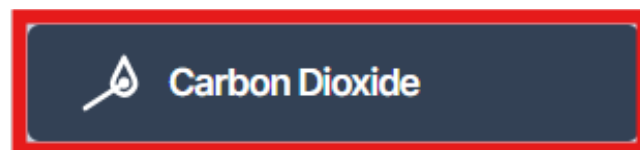
#### 2.1.1 Primary Energy

Esta opción corresponde a las gráficas en las que se tiene en cuenta como variable a las Energías Primarias, la energía primaria comprende los combustibles comercializados, incluidas las energías renovables modernas utilizadas para generar electricidad.



### ***2.1.2 Carbon Dioxide***

Esta opción corresponde a las gráficas en las que se tiene en cuenta como variable a las Emisiones de Dióxido de Carbono y la Quema de Gas Natural. Las emisiones de carbono mencionadas anteriormente reflejan únicamente aquellas provenientes del consumo de petróleo, gas y carbón para actividades relacionadas con la combustión, y se basan en 'Factores de emisiones de CO<sub>2</sub> predeterminados para la combustión' enumerados por el IPCC en sus Directrices para los inventarios nacionales de gases de efecto invernadero (2006). Esto no tiene en cuenta el carbono secuestrado, otras fuentes de emisiones de carbono ni las emisiones de otros gases de efecto invernadero.



### ***2.1.3 Oil***

Esta opción corresponde a las gráficas en las que se tiene en cuenta como variable a él Petróleo. Incluye petróleo crudo, petróleo de esquisto, arenas petrolíferas, condensados (condensados de arrendamiento o condensados de gas que requieren mayor refinación) y LGN (líquidos de gas natural: etano, GLP y nafta separados de la producción de gas natural). Excluye combustibles líquidos de otras fuentes como biocombustibles y derivados sintéticos del carbón y gas natural. Esto también excluye factores de ajuste del combustible líquido, como la ganancia en el procesamiento de la refinería. Excluye esquistos bituminosos/querógenos extraídos en forma sólida.



#### **2.1.4 Gas**

Esta opción corresponde a las gráficas en las que se tiene en cuenta como variable a el Gas. En la medida de lo posible, los datos anteriores representan metros cúbicos estándar (medidos a 15oC y 1013 mbar); como se derivan directamente de medidas de contenido de energía utilizando un factor de conversión promedio y se han estandarizado utilizando un poder calorífico bruto (GCV) de 40 MJ/m<sup>3</sup>, no necesariamente equivalen a volúmenes de gas expresados en términos nacionales específicos.



#### **2.1.5 Coal**

Esta opción corresponde a las gráficas en las que se tiene en cuenta como variable al Carbón.



#### **2.1.6 Nuclear Energy**

Esta opción corresponde a las gráficas en las que se tiene en cuenta como variable a la Energía Nuclear.



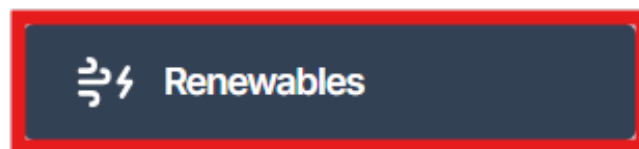
### ***2.1.7 Hydroelectricity***

Esta opción corresponde a las gráficas en las que se tiene en cuenta como variable a la Energía Hidráulica o Hidroelectricidad.



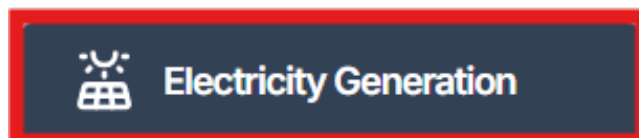
### ***2.1.8 Renewables***

Esta opción corresponde a las gráficas en las que se tiene en cuenta como variable a la Energías Renovables (Solar, Eólica, Geotérmica, Biomasa, Biocombustibles y otros). Incluye energía renovable (aparte de la hidráulica, que se informa por separado) y biocombustibles.



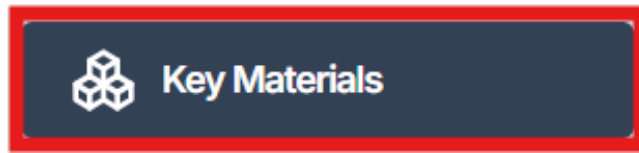
### ***2.1.9 Electricity Generation***

Esta opción corresponde a las gráficas en las que se tiene en cuenta como variable a la Generación de Energía Eléctrica, en esta generación se incluye el Petróleo, Gas, Carbón y otros.



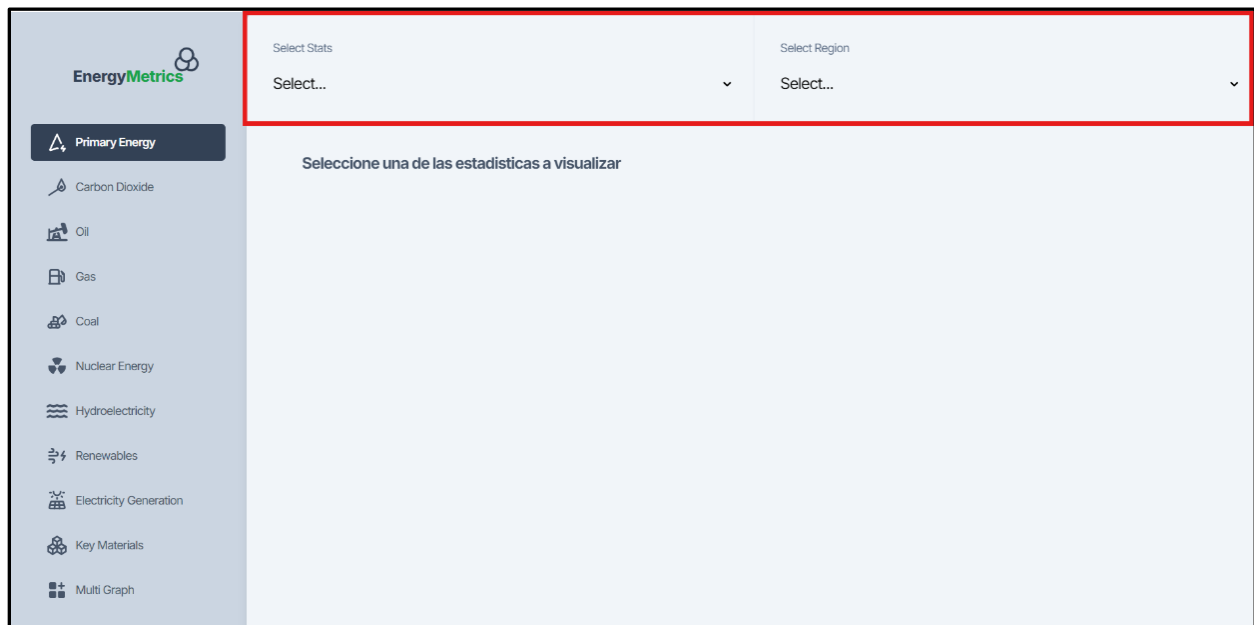
### ***2.1.10 Key Materials***

Esta opción corresponde a las gráficas en las que se tiene en cuenta como variable a los Materiales Claves en el desarrollo tecnológico y productivo tales como el Cobalto, Litio, Grafito y Tierras Raras.



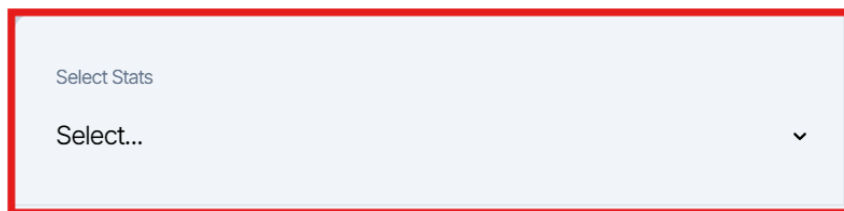
## 2.2 Filtros

En la sección de Filtros podemos ajustar las opciones para que las gráficas que requiramos se desplieguen de manera correcta en las unidades disponibles para cada variable.

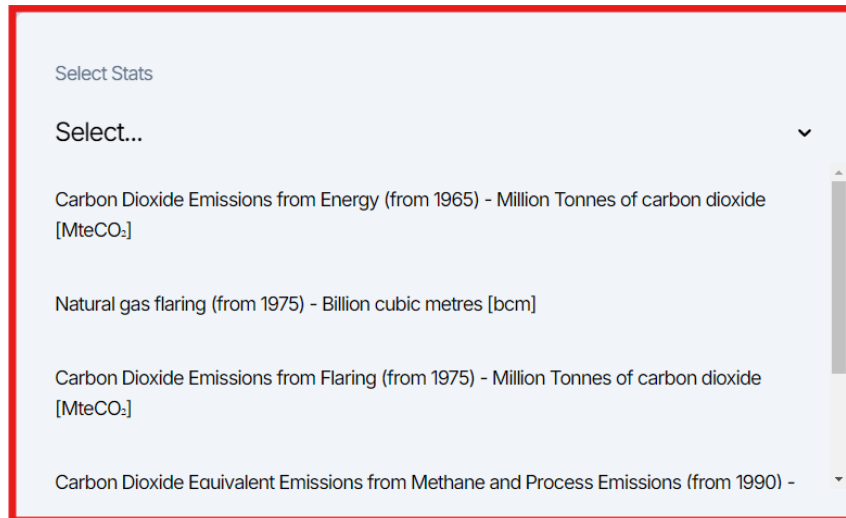


### 2.2.1 Stat Filter

En este filtro podemos seleccionar la estadística que deseemos, para desplegar el filtro es necesario hacer clic en cualquier espacio de este.

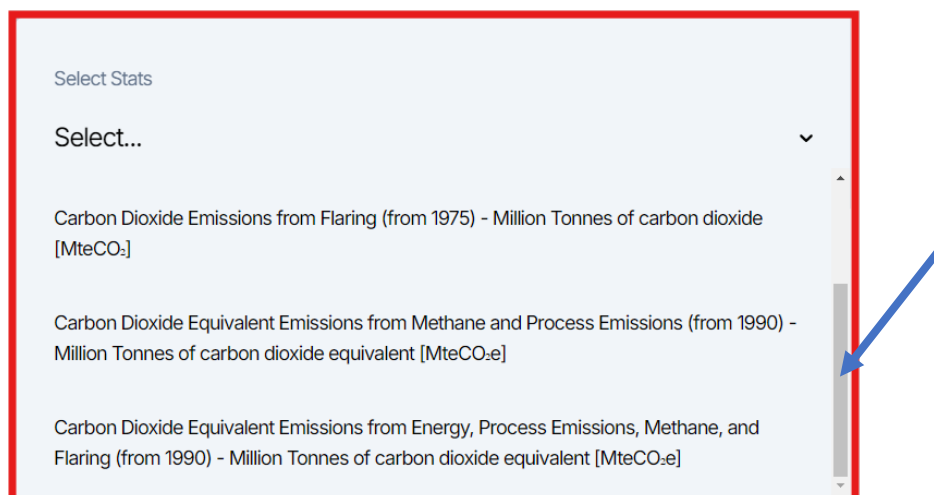


Después de desplegar el filtro vamos a encontrarnos con todas las opciones que incluye este filtro.

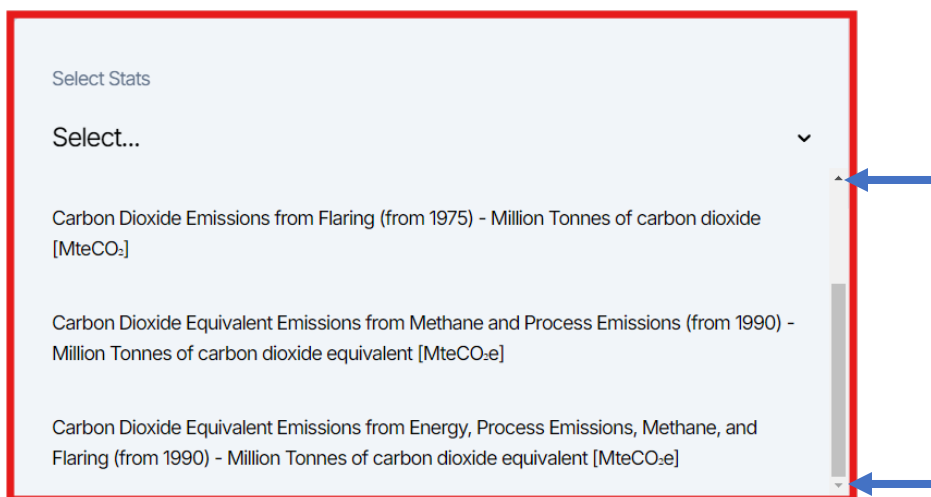


Si el filtro tiene más de cuatro opciones aparecerá una barra lateral en la cual podemos desplazarnos para ver todas las opciones. Para desplazarnos en esta barra es posible hacerlo por medio de 3 opciones (Cuando la barra llegue al final no se encontrarán más opciones):

- Utilizando la rueda del ratón
- Haciendo clic en la misma y arrastrándola hacia abajo



- Presionando las flechas subir y bajar que se encuentran en la barra



### 2.2.2 Region Filter

En este filtro podemos seleccionar la Región que deseemos aplicarle al Stat Filter, teniendo como opciones las siguientes:

- All Regions – Todas las Regiones
- North America – Norte América
- South & Central America – Sur y Centro América
- Europe – Europa
- Commonwealth of Independent States (CIS) – Comunidad de Estados Independientes (CEI)
- Middle East – Oriente Medio
- Africa – África
- Asia Pacific – Asia Pacifico

Para desplegar el filtro es necesario hacer clic en cualquier espacio de este.

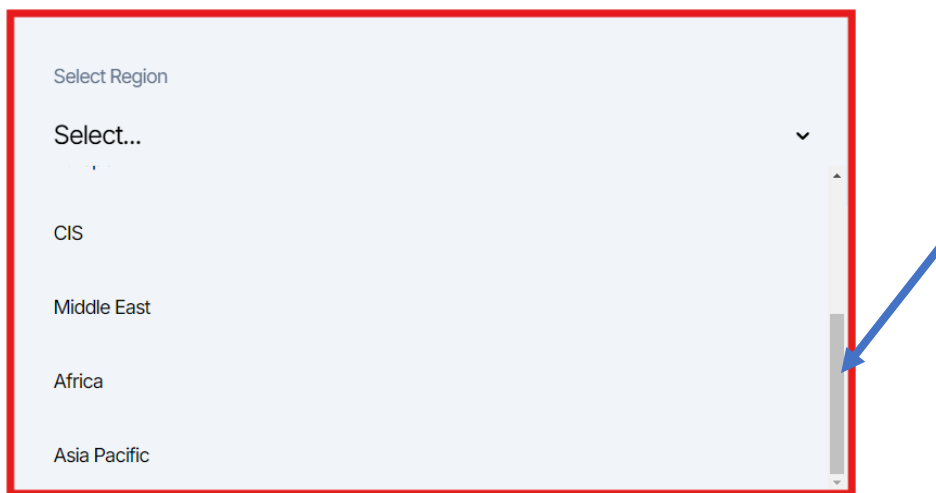


Después de desplegar el filtro vamos a encontrarnos con todas las opciones que incluye este filtro.

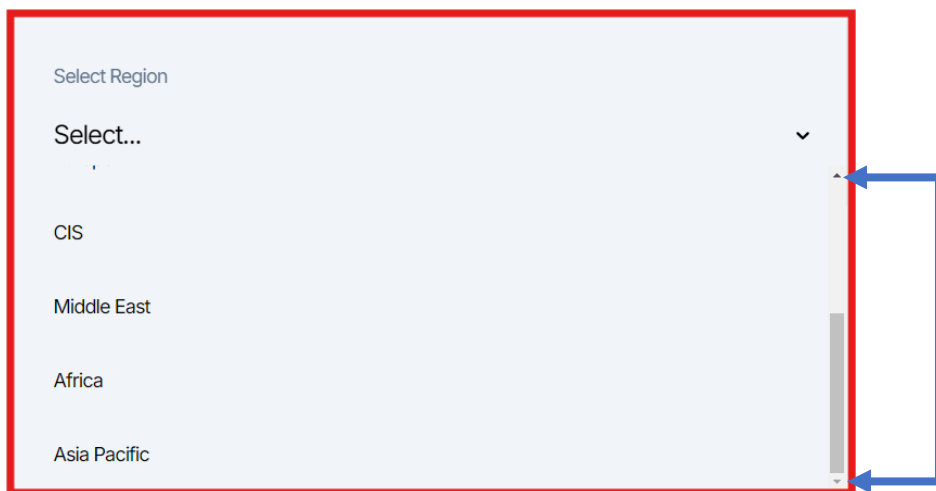


Si el filtro tiene más de cuatro opciones aparecerá una barra lateral en la cual podemos desplazarnos para ver todas las opciones. Para desplazarnos en esta barra es posible hacerlo por medio de 3 opciones (Cuando la barra llegue al final no se encontrarán más opciones):

- Utilizando la rueda del ratón
- Haciendo clic en la misma y arrastrándola hacia abajo



- Presionando las flechas subir y bajar que se encuentran en la barra

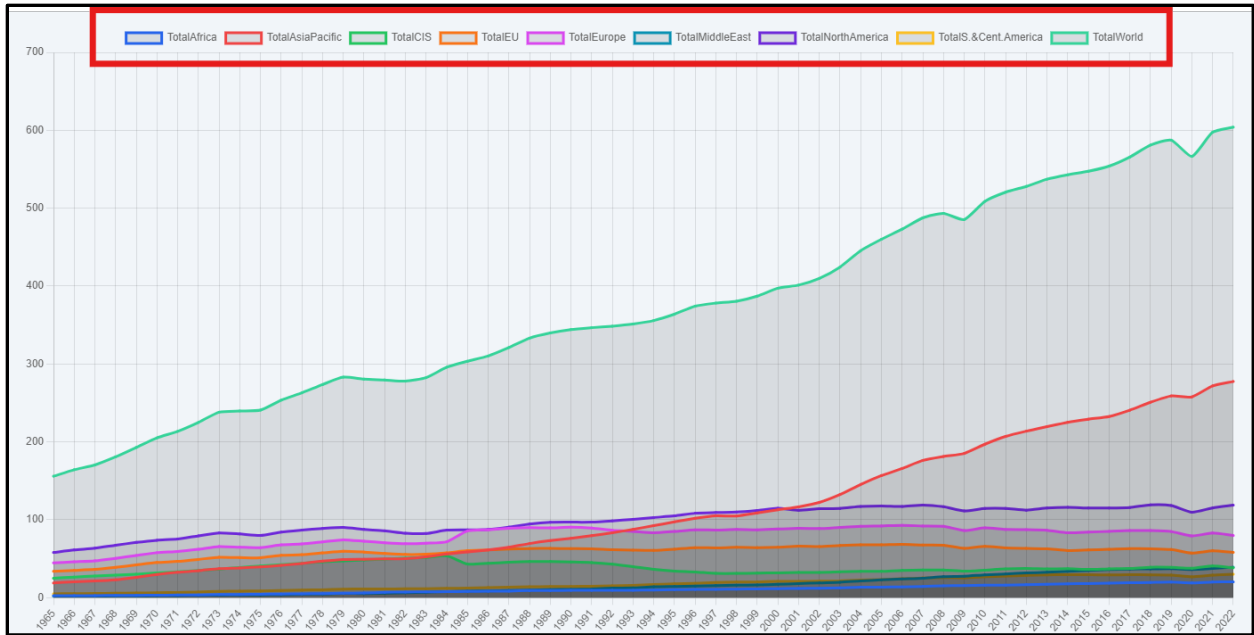


### ***2.2.3 Graph Region/Country Filter***

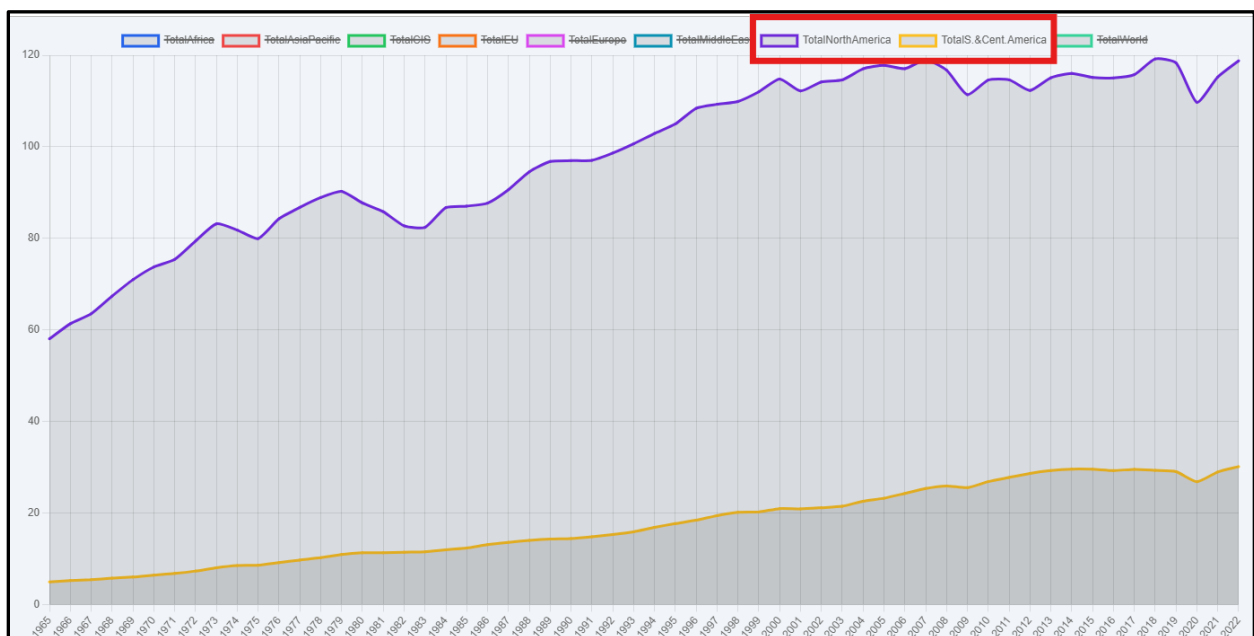
Este filtro lo podremos encontrar en cada una de las gráficas ya sea para eliminar regiones si tenemos la opción de todas las regiones seleccionada en el filtro de región o eliminar países en el caso que tengamos una región en específico seleccionada en el filtro de región.

#### ***2.2.3.1 Graph Region Filter***

Si en el filtro de región seleccionamos todas las regiones el programa nos entregara una gráfica similar a la siguiente en la que encontraremos todas las regiones en la parte superior de esta:



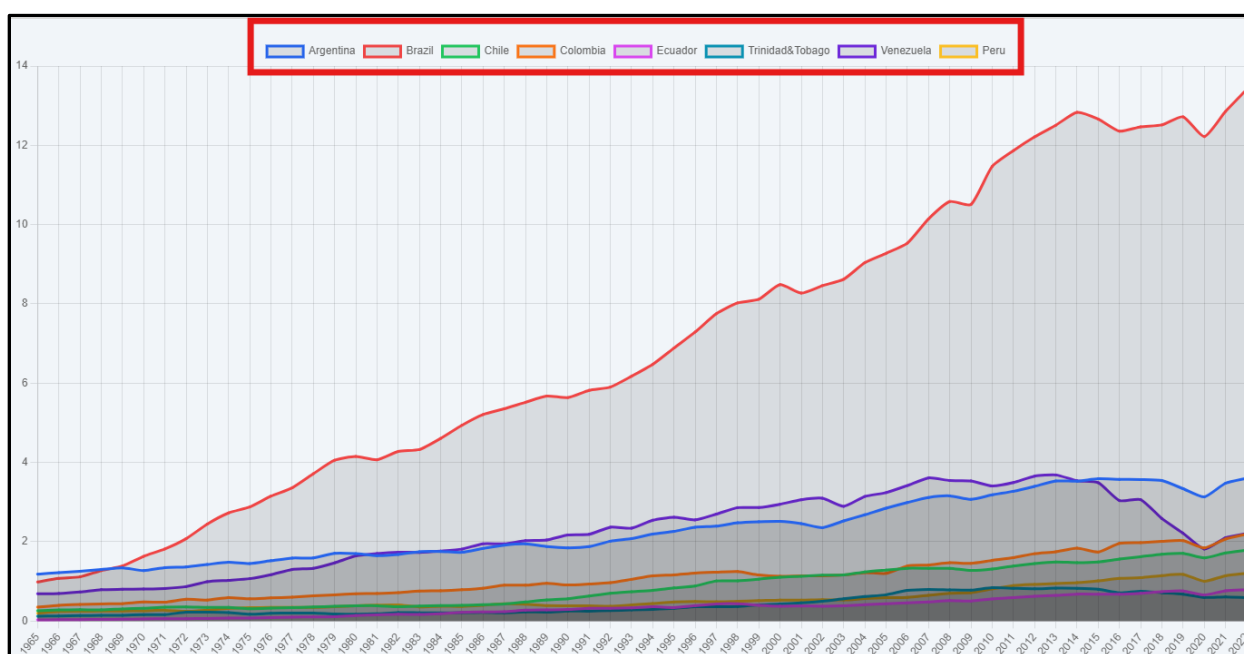
En este filtro podremos eliminar todas las regiones que deseemos simplemente haciendo clic sobre ellas e inmediatamente la gráfica se actualizara con nuestras necesidades



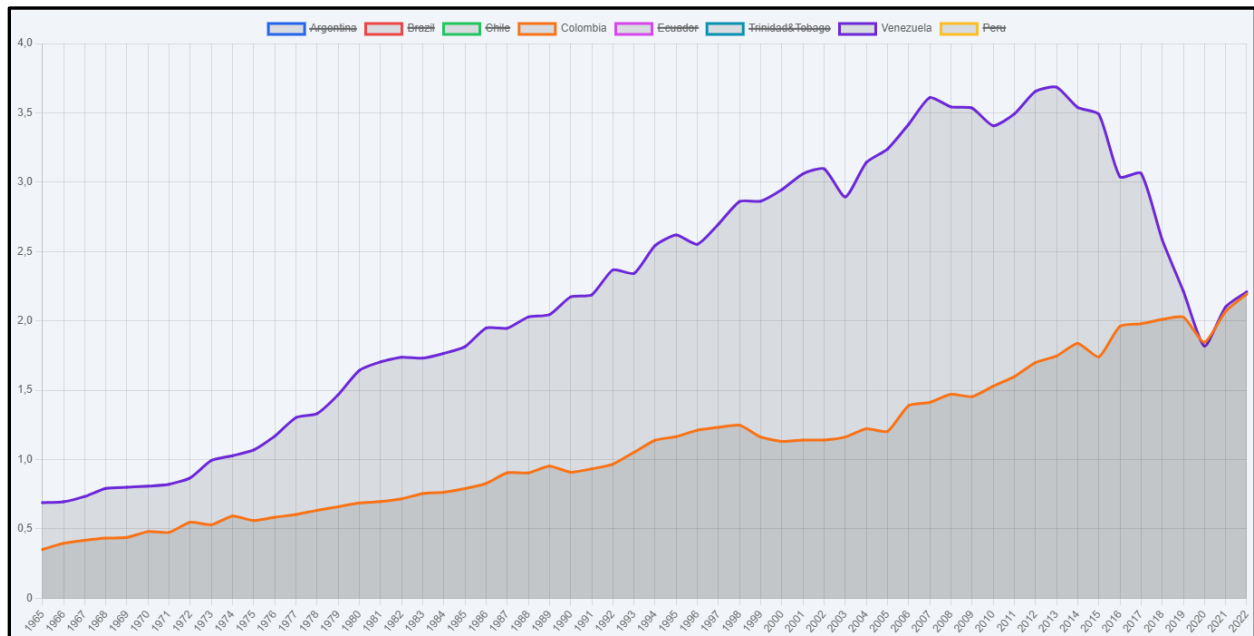
En el ejemplo de la imagen anterior podemos visualizar que se eliminaron todas las regiones y solo se dejaron Norte América y Sur/Centro América.

### 2.2.3.2 Graph Country Filter

Si en el filtro de región seleccionamos alguna región en concreto el programa nos entregara una gráfica similar a la siguiente en la que encontraremos todas los países que comprenden a esa región en la parte superior de esta:



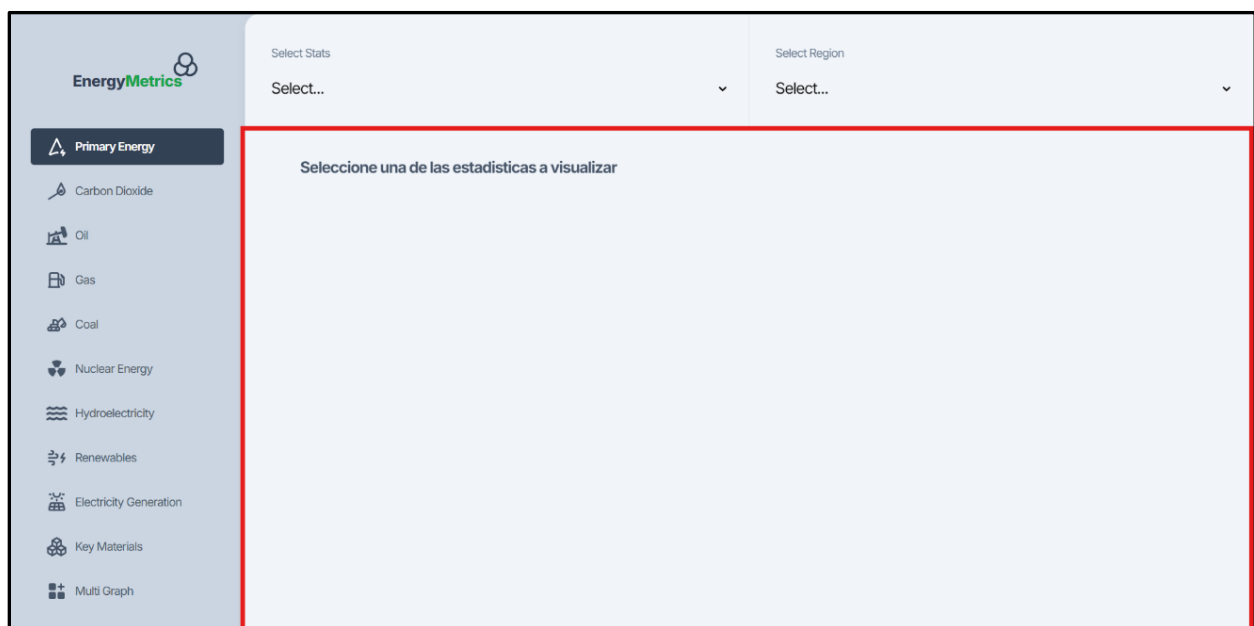
En este filtro podremos eliminar todos los países que deseemos simplemente haciendo clic sobre ellos e inmediatamente la gráfica se actualizara con nuestras necesidades



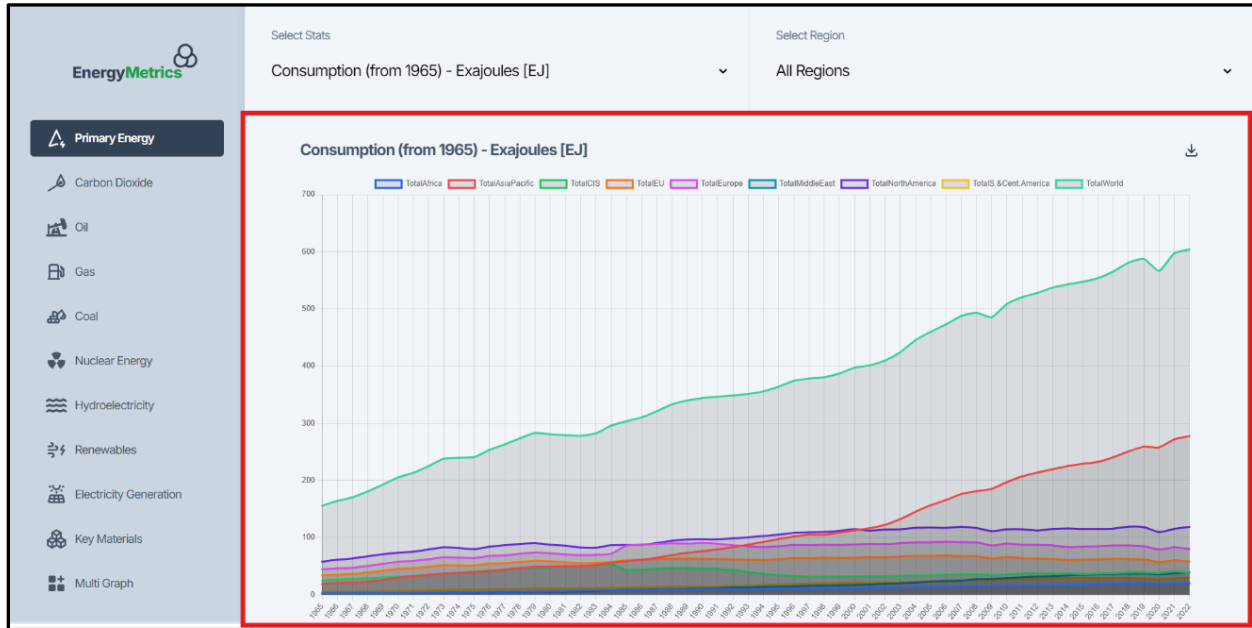
En el ejemplo de la imagen anterior podemos visualizar que se eliminaron todos los países y solo se dejaron Colombia y Venezuela.

### 2.3 Display Zone

En esta zona nos encontraremos con la gráfica, en caso de que aún no seleccionemos los filtros correspondientes nos encontraremos con una zona vacía

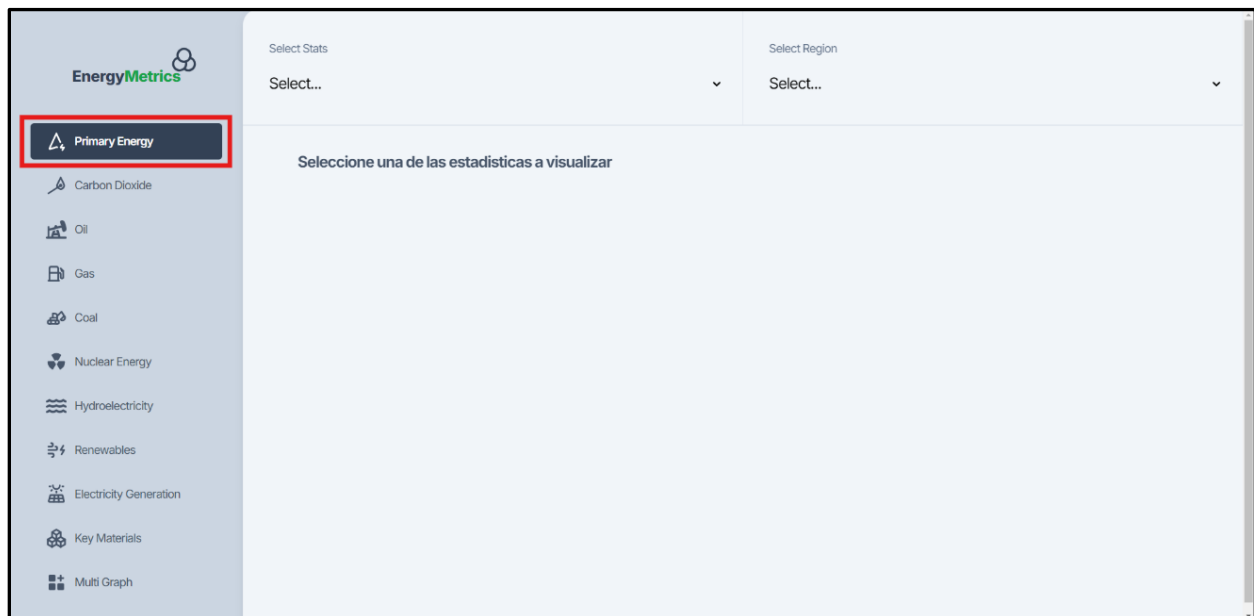


Después de seleccionar los filtros nuestra grafica se mostrará en ese espacio que anteriormente se encontraba vacío

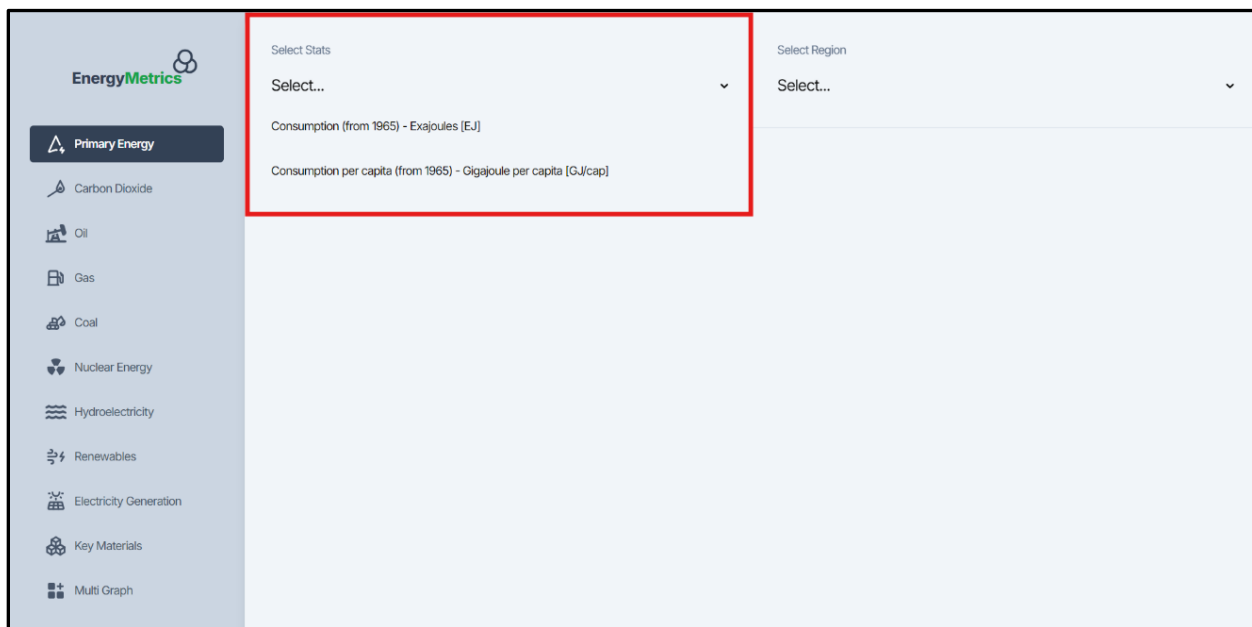


### 3 Como Graficar

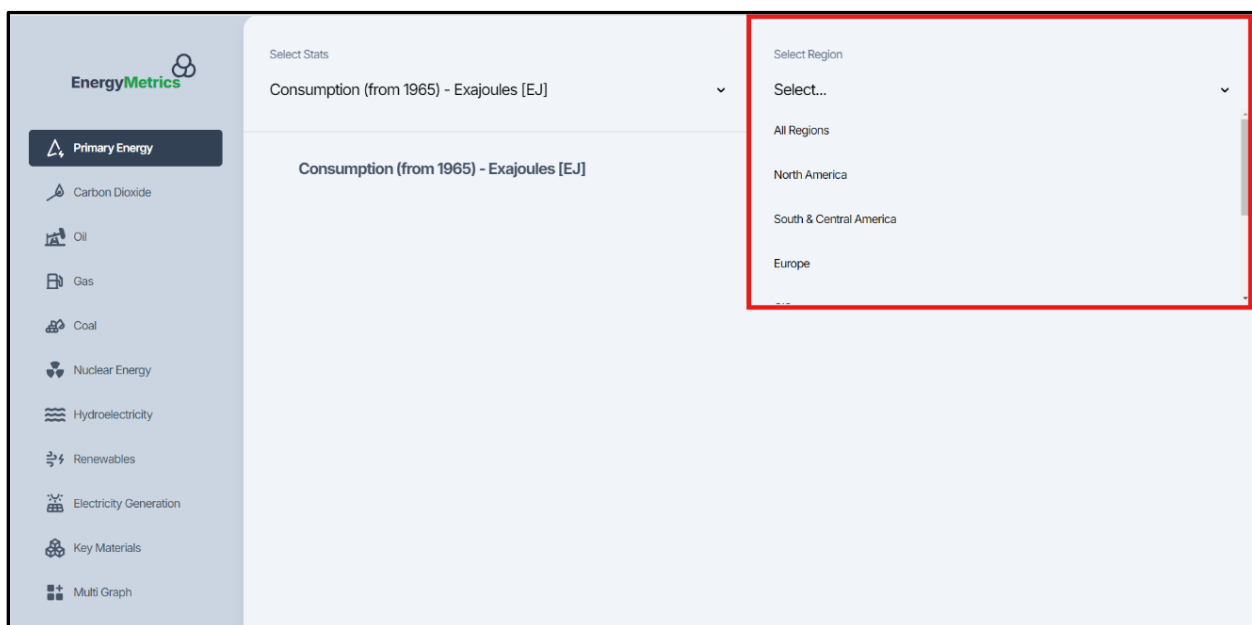
Seleccionar una de las opciones del Side Nav expuestas en el numeral 2.1



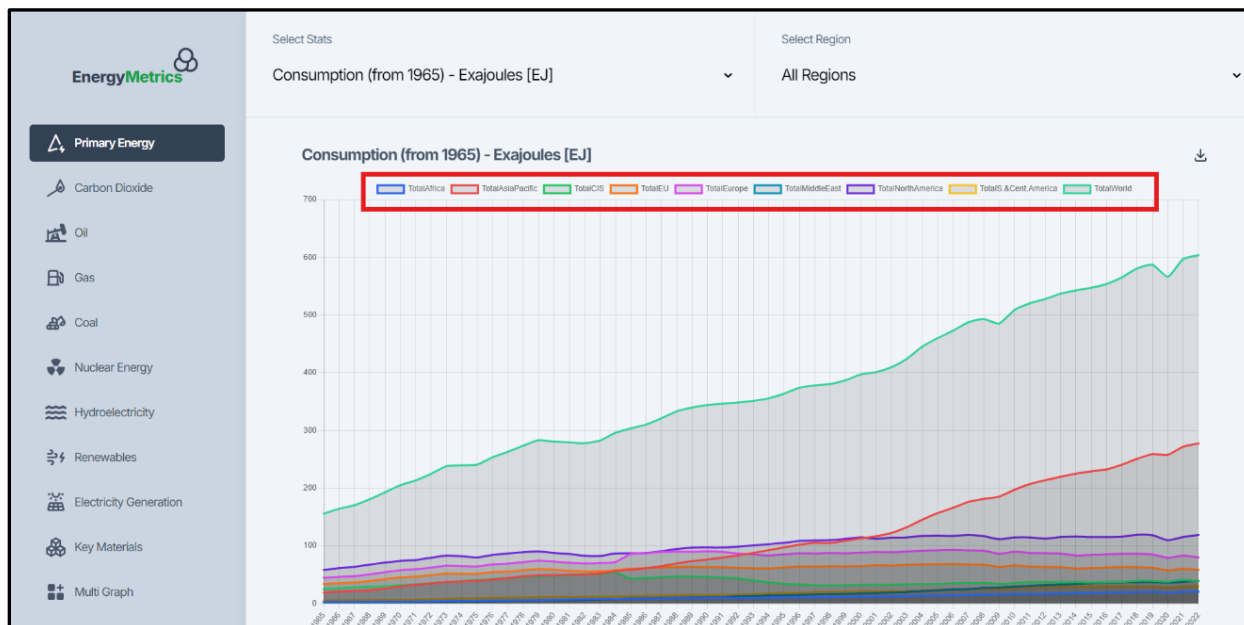
Seleccionar una de las opciones del Stat Filter expuestas en el numeral 2.2.1



Seleccione una de las opciones del Region Filter expuestas en el numeral 2.2.2



Seleccione los países o regiones que desea eliminar de la gráfica expuestos en el Graph Region/Country Filter expuesto en el numeral 2.2.4



Después de seleccionar las que desee eliminar estas se van a ver con una línea gris tachándolas



Para volver a incluirlas en la gráfica simplemente vuelva a dar clic sobre las opciones que aparecen con la línea gris sobre ellas



### 3.5 Capítulo 5 – Conclusiones

Después de realizar un análisis exhaustivo de los resultados obtenidos tras la implementación de la aplicación web diseñada para el almacenamiento y análisis de datos en la industria petrolera. Este análisis destaca la eficacia sobresaliente del sistema, especialmente

gracias a la implementación de MongoDB como sistema de gestión de bases de datos. MongoDB ha demostrado ser excepcionalmente eficaz en la gestión de grandes volúmenes de datos complejos y no estructurados, lo cual es esencial en el sector petrolero. Su modelo de datos basado en documentos permite una flexibilidad y escalabilidad que son cruciales para adaptarse a las dinámicas cambiantes de la industria, facilitando un acceso rápido y fiable a la información necesaria para la toma de decisiones estratégicas.

Más allá de su impacto en el sector petrolero, la aplicación también aporta beneficios significativos al ámbito educativo. La integración de la base de datos BP Statistical Review 2022 ha enriquecido la aplicación con datos históricos y contextuales esenciales para la formación de ingenieros de petróleo. Estos datos permiten a los estudiantes y académicos explorar la evolución de las energías y entender los cambios en el panorama energético global a lo largo del tiempo. Esto no solo mejora la calidad de la educación, sino que también prepara a los futuros profesionales para enfrentar los desafíos del sector con una comprensión más profunda de su historia y tendencias actuales.

La aplicación ha demostrado ser una herramienta valiosa para la investigación y la formación, permitiendo a los estudiantes trabajar con datos reales del sector. Esto facilita el desarrollo de proyectos de investigación más relevantes y proporciona una plataforma práctica para aplicar teorías aprendidas en el aula. La capacidad de la aplicación para procesar y visualizar datos complejos de manera intuitiva permitirá a los educadores integrar estos recursos en sus planes de estudio, mejorando la experiencia de aprendizaje y fomentando un enfoque más analítico y basado en datos.

En resumen, los resultados obtenidos subrayan el impacto positivo de la aplicación no solo en la industria petrolera, sino también en el sector educativo. La combinación de MongoDB, Next.js, Visual Studio Code y la base de datos BP Statistical Review 2022 ha proporcionado una infraestructura robusta y eficiente que impulsa la innovación y la eficiencia, demostrando cómo la tecnología puede transformar y mejorar sectores tradicionales. Estos logros destacan la importancia de integrar herramientas tecnológicas avanzadas en la educación y la industria para fomentar un desarrollo sostenible y basado en datos.

### **3.6 Capítulo 6 – Recomendaciones y Futuras Mejoras**

En este capítulo se analiza las estrategias y desarrollos propuestos para optimizar y expandir la funcionalidad de la aplicación web en el futuro. Estas recomendaciones se basan en la experiencia adquirida durante el desarrollo inicial y en las necesidades emergentes identificadas en el sector petrolero y educativo.

Una de las principales mejoras sugeridas es la implementación de un módulo administrativo que permita a los usuarios acceder y gestionar los datos directamente desde la aplicación web. Este módulo proporcionaría una interfaz intuitiva para la entrada de datos, facilitando la actualización y gestión de la información sin necesidad de intervención técnica directa. Dentro de este módulo, se propone crear una subfunción que permita actualizar la base de datos de manera más eficiente, evitando la necesidad de normalizar los datos manualmente y reduciendo así el tiempo y esfuerzo requeridos para mantener la integridad de la base de datos.

Otra recomendación clave es realizar un estudio con Google Ads para integrar publicidad en la página web. Esta estrategia no solo ayudaría a financiar futuros desarrollos y mantenimientos

de la aplicación, sino que también podría aumentar la visibilidad y el alcance de la aplicación dentro del sector petrolero y más allá.

Además, se sugiere la creación de un módulo de usuarios que permita recolectar información sobre los diferentes tipos de usuarios que interactúan con la aplicación. Este módulo no solo facilitaría la personalización de la experiencia del usuario, sino que también proporcionaría datos valiosos para generar análisis y reportes sobre el comportamiento y las necesidades de los usuarios.

Para enriquecer aún más la base de datos, se recomienda integrar otras bases de datos relacionadas con la industria petrolera. Esto ampliaría el alcance de los análisis posibles y proporcionaría una visión más completa y detallada del sector, beneficiando tanto a los profesionales como a los estudiantes que utilizan la aplicación como herramienta educativa.

La integración de inteligencia artificial en la aplicación es otra mejora significativa propuesta. La IA podría utilizarse para realizar análisis de datos avanzados, identificar patrones y tendencias ocultas, y proporcionar recomendaciones basadas en datos históricos y en tiempo real. Esto no solo mejoraría la capacidad de la aplicación para ofrecer insights valiosos, sino que también aumentaría su utilidad para la toma de decisiones estratégicas.

Finalmente, se propone el desarrollo de un módulo empresarial que permita almacenar y visualizar datos en tiempo real de los pozos petroleros. Este módulo proporcionaría a las empresas una herramienta poderosa para monitorear sus operaciones y optimizar la producción, asegurando que la aplicación siga siendo relevante y valiosa en un entorno empresarial competitivo.

Estas recomendaciones y mejoras futuras están diseñadas para asegurar que la aplicación no solo mantenga su relevancia, sino que también continúe evolucionando para satisfacer las necesidades cambiantes de la industria petrolera y educativa.

Para respaldar las recomendaciones propuestas, es útil considerar ejemplos de éxito de otros proyectos que han implementado características similares con resultados positivos. Aunque los resultados de búsqueda no proporcionan ejemplos específicos, podemos inferir prácticas comunes y exitosas en el desarrollo de aplicaciones web y sistemas de gestión de datos.

***Módulo Administrativo y Actualización de Datos:*** Muchas aplicaciones exitosas han implementado módulos administrativos que permiten a los usuarios gestionar y actualizar datos de manera eficiente. Por ejemplo, sistemas de gestión de contenido como WordPress y Drupal ofrecen interfaces administrativas intuitivas que permiten a los usuarios actualizar contenido sin necesidad de conocimientos técnicos avanzados.

***Publicidad y Monetización:*** El uso de Google Ads para monetizar aplicaciones web es una práctica común que ha demostrado ser efectiva en numerosos proyectos. Plataformas como YouTube y blogs populares utilizan Google Ads para generar ingresos pasivos que financian mejoras continuas y mantenimiento.

***Módulo de Usuarios y Analíticas:*** Aplicaciones como Facebook y LinkedIn han implementado módulos de usuarios que no solo mejoran la experiencia del usuario, sino que también permiten recolectar datos valiosos para análisis. Estos datos pueden ser utilizados para personalizar la experiencia del usuario y mejorar la toma de decisiones estratégicas.

***Integración de Inteligencia Artificial:*** La integración de inteligencia artificial para el análisis de datos es una tendencia creciente en la industria. Empresas como Netflix y Amazon utilizan algoritmos de IA para analizar grandes volúmenes de datos y proporcionar recomendaciones personalizadas a sus usuarios.

***Módulo Empresarial para Datos en Tiempo Real:*** La capacidad de visualizar datos en tiempo real es crucial en industrias donde la toma de decisiones rápida es esencial. Aplicaciones como Tableau y Power BI permiten a las empresas visualizar datos en tiempo real, mejorando la eficiencia operativa y la capacidad de respuesta.

Estos ejemplos de éxito destacan la importancia de implementar mejoras tecnológicas que no solo optimicen la funcionalidad de la aplicación, sino que también proporcionen valor añadido a los usuarios y stakeholders. Al aprender de estos casos, la tesis puede proponer mejoras que aseguren la relevancia y sostenibilidad a largo plazo de la aplicación en el sector petrolero y educativo.

### **3.7 Capítulo 7 – Unidades y Factores de Conversión Aproximados**

#### **3.7.1 Unidades**

##### **Exajoules [EJ]**

Un “Exajulio” es una unidad de medida de energía en el Sistema Internacional de Unidades (SI). Es una unidad extremadamente grande, utilizada principalmente para expresar cantidades significativas de energía, como el consumo energético de países enteros o la producción de energía a nivel global. Debido a su magnitud, el Exajulio es especialmente útil en informes y estudios energéticos que involucran grandes escalas, como el análisis de las reservas de energía mundial o el impacto energético de grandes infraestructuras.

**Gigajoule per capita [GJ/cap]**

La unidad de medida “Gigajulio per cápita” se refiere a la cantidad de energía, medida en Gigajulios, consumida o producida por cada persona en un área o país específico. Un Gigajulio es una unidad de energía en el Sistema Internacional de Unidades (SI). Esta medida es útil para comparar el consumo o producción de energía entre diferentes poblaciones, proporcionando un contexto sobre la eficiencia energética o el nivel de desarrollo energético de una región.

**Million Tonnes of carbon dioxide [MteCO<sub>2</sub>]**

La unidad de medida “Millones de toneladas de dióxido de carbono” se utiliza para cuantificar las emisiones de dióxido de carbono (CO<sub>2</sub>) en términos de millones de toneladas métricas. Esta medida es comúnmente empleada en informes ambientales y estudios sobre cambio climático para evaluar el impacto de diferentes actividades humanas, como la quema de combustibles fósiles, en la concentración de CO<sub>2</sub> en la atmósfera.

**Billion cubic meters [bcm]**

La unidad de medida “Mil millones de metros cúbicos” se refiere a mil millones de metros cúbicos y es comúnmente utilizada para cuantificar grandes volúmenes de gas natural. Esta unidad es especialmente relevante en la industria energética, donde se emplea para medir la producción, consumo, y reservas de gas natural a nivel nacional e internacional.

**Million Tonnes of carbon dioxide equivalent [MteCO<sub>2</sub>e]**

La unidad de medida “Millones de toneladas de dióxido de carbono equivalente” se utiliza para cuantificar las emisiones de gases de efecto invernadero (GEI) en términos de su equivalente en dióxido de carbono (CO<sub>2</sub>). Esta medida permite expresar la cantidad total de diferentes GEI en una sola cifra, utilizando el CO<sub>2</sub> como referencia común. Esto se logra multiplicando la cantidad

de cada gas por su potencial de calentamiento global (GWP), que es una medida de cuánto contribuye un gas al calentamiento global en comparación con el CO<sub>2</sub>.

### **Thousand barrels per day [Kb/d]**

La unidad de medida “Mil barriles por día” se utiliza para expresar la producción o el consumo de petróleo en miles de barriles por día. Esta medida es comúnmente empleada en la industria petrolera para cuantificar la capacidad de producción de pozos, refinerías, o incluso la demanda de petróleo en regiones o países específicos.

### **Million tonnes [Mte]**

La unidad de medida “Millones de toneladas” se utiliza para expresar grandes cantidades de masa, generalmente en el contexto de recursos naturales, productos agrícolas, o emisiones de gases. Un millón de toneladas métricas es una cantidad significativa, y esta unidad es comúnmente utilizada en informes y análisis que involucran grandes volúmenes de materiales o sustancias.

### **Billion cubic feet per day [bcf/d]**

La unidad de medida “Mil millones de pies cúbicos por día” se utiliza para cuantificar el flujo de gas natural, indicando específicamente cuántos mil millones de pies cúbicos de gas se producen, consumen o transportan en un día. Esta unidad es comúnmente utilizada en la industria energética para expresar grandes volúmenes de gas natural, facilitando la comparación y comunicación de datos relacionados con la producción y consumo de gas a gran escala.

### **Terawatt hour [Twh]**

La unidad de medida “Terawatt-hora” es una unidad de energía que equivale a un billón ( $10^{12}$ ) de vatios-hora. Se utiliza comúnmente para expresar grandes cantidades de energía, como

el consumo anual de electricidad de un país o la producción de energía de una planta eléctrica a gran escala.

### **Thousand barrels of oil equivalent per day [Kboe/d]**

La unidad de medida "Mil barriles de petróleo equivalente por día" se utiliza para expresar la producción o consumo de energía en términos de barriles de petróleo equivalente. Esta medida es comúnmente empleada en la industria energética para cuantificar la producción de petróleo y gas, así como para comparar diferentes fuentes de energía en una base común.

### **Petajoules [PJ]**

La unidad de medida "Petajulios" es una unidad de energía en el Sistema Internacional de Unidades (SI). Se utiliza para expresar grandes cantidades de energía, como el consumo energético de sectores industriales o la producción de energía de grandes infraestructuras energéticas, debido a su gran magnitud.

### **Thousand tonnes [kt]**

La unidad de medida "Mil toneladas" se utiliza para expresar grandes cantidades de masa, generalmente en el contexto de recursos naturales, productos agrícolas o emisiones de gases. Esta unidad es comúnmente empleada en informes y análisis que involucran grandes volúmenes de materiales o sustancias.

### **Million Tonnes of Oil Equivalent [Mtoe]**

La unidad de medida "Millones de toneladas de petróleo equivalente" se utiliza para expresar grandes cantidades de energía, comparando diferentes fuentes de energía en términos de su contenido energético equivalente al del petróleo. Una tonelada de petróleo equivalente es una

unidad estándar de energía que representa la cantidad de energía liberada por la combustión de una tonelada de petróleo crudo.

**Trillion British Thermal Units [Tbtu]**

La unidad de medida “Billón de unidades térmicas británicas” se utiliza para expresar grandes cantidades de energía, particularmente en el contexto del gas natural y otras fuentes de energía. Un British Thermal Unit (BTU) es una unidad de energía que se define como la cantidad de energía necesaria para elevar la temperatura de una libra de agua en un grado Fahrenheit.

**Million Barrels of Oil Equivalent [Mboe]**

La unidad de medida “Millones de barriles de petróleo equivalente” se utiliza para expresar grandes cantidades de energía, comparando diferentes fuentes de energía en términos de su contenido energético equivalente al del petróleo. Un barril de petróleo equivalente es una unidad estándar que representa la cantidad de energía contenida en un barril de petróleo crudo.

**3.7.2 Factores de Conversión Aproximados**

**Crude Oil**

**Tabla 1 – Factores de Conversión del Petróleo**

	To				
	Tonnes (metric)	Kilolitres	Barrels	US gallons	Tonnes/year
From	Multiply by				
Tonnes (metric)	1	1.165	7.33	307.86	-
Kilolitres	0.8581	1	6.2898	264.17	-
Barrels	0.1364	0.159	1	42	-
US gallons	0.00325	0.0038	0.0238	1	-
Barrels/day	-	-	-	-	49.8

Nota. Esta tabla muestra los factores de conversión aproximados para el petróleo basado en la gravedad promedio mundial. Tomado de “BP Stats Review 2022 All Data”, 2022.

**Oil Products**

**Tabla 2 – Factores de Conversión de Subproductos del Petróleo**

From	To					
	barrels to tonnes	tonnes to barrels	kiloliters to tonnes	tonnes to kiloliters	tonnes to gigajoules	tonnes to barrels oil equiv.
	<b>Multiply by</b>					
Ethane	0.059	16.850	0.373	2.679	49.400	8.073
Liquefied petroleum gas (LPG)	0.086	11.600	0.541	1.849	46.150	7.542
Gasoline	0.120	8.350	0.753	1.328	44.750	7.313
Kerosene	0.127	7.880	0.798	1.253	43.920	7.177
Gas oil / diesel	0.134	7.460	0.843	1.186	43.380	7.089
Residual fuel oil	0.157	6.350	0.991	1.010	41.570	6.793
Product basket	0.124	8.058	0.781	1.281	43.076	7.039

Nota. Esta tabla muestra los factores de conversión aproximado para los subproductos del petróleo. Tomado de “BP Stats Review 2022 All Data”, 2022.

**Natural gas (NG) and liquefied natural gas (LNG)**

**Tabla 3 – Factores de Conversión del Gas Natural y el Gas Natural Licuad**

From	To						
	bcm NG	bcf NG	PJ NG	Mtoe	Mte LNG	Tbtu	Mboe
	<b>Multiply by</b>						
1 bcm NG	1.000	35.315	36.000	0.860	0.735	34.121	5.883
1 bcf NG	0.028	1.000	1.019	0.024	0.021	0.966	0.167
1 PJ NG	0.028	0.981	1.000	0.024	0.021	0.952	0.164
1 Mtoe	1.163	41.071	41.868	1.000	0.855	39.683	6.842
1 Mte LNG	1.360	48.028	48.747	1.169	1.000	46.405	8.001
1 Tbtu	0.029	1.035	1.050	0.025	0.022	1.000	0.172
1 Mboe	0.170	6.003	6.093	0.146	0.125	5.800	1.000

Nota. Esta tabla muestra los factores de conversión aproximado para el gas natural y el gas natural licuado. Tomado de “BP Stats Review 2022 All Data”, 2022.

**Thermal equivalent efficiency factors used to convert non-fossil electricity to primary energy**

**Tabla 4 – Factores de Eficiencia**

Year(s)	Efficiency factor
1965-2000	36%
2001	36.20%
2002	36.50%
2003	36.70%
2004	36.90%
2005	37.20%
2006	37.40%
2007	37.60%
2008	37.90%
2009	38.10%
2010	38.40%
2011	38.60%
2012	38.80%
2013	39.10%
2014	39.30%
2015	39.50%
2016	39.80%
2017	40.00%
2018	40.20%
2019	40.40%
2020	40.50%
2021	40.60%

Nota. Esta tabla muestra los factores de eficiencia equivalentes térmicos utilizados para convertir la electricidad no fósil en energía primaria. Tomado de “BP Stats Review 2022 All Data”, 2022.

**Factores de Conversión Aproximados para Unidades**

- 1 metric tonne = 2204.62 lb = 1.1023 short tons
- 1 kilolitre = 6.2898 barrels = 1 cubic metre
- 1 kilocalorie [kcal] = 4.1868 kJ = 3.968 Btu
- 1 kilojoule [kJ] = 1,000 joules = 0.239 kcal = 0.948 Btu
- 1 petajoule [PJ] = 1 quadrillion joules (1 x 10<sup>15</sup>)
- 1 exajoule [EJ] = 1 quintillion joules (1 x 10<sup>18</sup>)
- 1 British thermal unit [Btu] = 0.252 kcal = 1.055 kJ
- 1 barrel of oil equivalent [boe] = 5.8 million Btu = 6.119 million kJ
- 1 kilowatt – hour (kWh) = 860 kcal = 3600 kJ = 3412 Btu

**Equivalentes Caloríficos**

Un Exajulio equivale aproximadamente a:

**Tabla 5 – Equivalencias Caloríficas de un Exajulio**

Heat units	239 trillion kilocalories
	948 trillion Btu
Solid fuels	40 million tonnes of hard coal
	95 million tonnes of lignite and sub-bituminous coal
Gaseous fuels	See Natural gas and LNG table
Electricity	278 terawatt-hours

Nota. Esta tabla muestra las equivalencias caloríficas de un Exajulio en unidades de calor, combustibles sólidos, combustibles gaseosos y electricidad. Tomado de “BP Stats Review 2022 All Data”, 2022.

**Valores Caloríficos Netos o Inferiores**

- *1 barrel of ethanol = 0.58 barrels of oil equivalent*
- *1 barrel of biodiesel = 0.86 barrels of oil equivalent*
- *1 tonne of ethanol = 0.68 tonne of oil equivalent*
- *1 tonne of biodiesel = 0.88 tonne of oil equivalent*

### Referencias Bibliográficas

- Abadi, D. (2012). Consistency tradeoffs in modern distributed database system design: CAP is only part of the story. *Computer*, 45(2), 37-42.
- Accomazzo, A., Murray, N., Lerner, A. (2017). *Fullstack React: The Complete Guide to ReactJS and Friends*. Estados Unidos: Fullstack.io.
- Adobe. (2023). Adobe Dreamweaver. Obtenido de <https://www.adobe.com/products/dreamweaver.html>
- Albahari, J., & Albahari, B. (2020). *C# 8.0 in a Nutshell: The Definitive Reference*. O'Reilly Media.
- Allen, G., & Owens, M. (2010). *The Definitive Guide to SQLite*. Apress.
- Apache Software Foundation. (2023). Apache Cassandra. Obtenido de <https://cassandra.apache.org/>
- Apache Software Foundation. (2023). Apache NetBeans. <https://netbeans.apache.org/>
- Arnold, K., Gosling, J., & Holmes, D. (2005). *The Java Programming Language (4th ed.)*. Addison-Wesley Professional.
- Artech. (2023). GeneXus: The Low-Code Platform for Enterprise Applications. Obtenido de <https://www.genexus.com/>
- Atom. (2023). Atom Features. <https://atom.io/>
- Axel, R. (2016). *Exploring ES6: Upgrade to the next version of JavaScript*. Leanpub.
- Banker, K. (2011). *MongoDB in Action*. Manning Publications.
- Bartholomew, D. (2015). *MariaDB Cookbook*. Packt Publishing Ltd.
- Beazley, D. M. (2009). *Python Essential Reference (4th ed.)*. Addison-Wesley Professional.

- Bell, C., Kindahl, M., & Thalmann, L. (2014). *MySQL High Availability: Tools for Building Robust Data Centers*. O'Reilly Media, Inc.
- Ben-Gan, I. (2020). *T-SQL Fundamentals (4th ed.)*. Microsoft Press.
- Betancourt, D. F. (26 de Julio de 2016). *Ingenio Empresa*. Obtenido de <https://www.ingenioempresa.com/diagrama-de-dispersion/>
- Beynon-Davies, P. (2019). *Sistemas de bases de datos*. REVERTE
- Biessek, A. (2019). *Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2*. Packt Publishing.
- Bloch, J. (2018). *Effective Java (3rd ed.)*. Addison-Wesley Professional.
- Böck, H. (2011). *The Definitive Guide to NetBeans Platform 7*. Apress.
- Bonnie, E. (2017). *React Native - Building Mobile Apps with JavaScript*. Packt Publishing.
- Boudreau, T., Tulach, J., & Wielenga, G. (2015). *Rich Client Programming: Plugging into the NetBeans Platform*. Prentice Hall.
- BP. (2022). *Statistical Review of World Energy*. <https://www.bp.com/en/global/corporate/energy-economics/statistical-review-of-world-energy.html>
- Bucă, A. (2020). *Next.js Quick Start Guide: Server-side rendering done right*. Packt Publishing.
- Buchanan, M. (2020). *Ionic Framework by Example*. Packt Publishing.
- Burnette, E. (2005). *Eclipse IDE Pocket Guide*. O'Reilly Media.
- Cairo, A. (2013). *The Functional Art: An introduction to information graphics and visualization*. New Riders.
- Cairo, A. (2016). *The Truthful Art: Data, Charts, and Maps for Communication*. New Riders.
- Cantu, M. (2021). *Delphi Cookbook (4th ed.)*. Packt Publishing.

- Carpenter, J., & Hewitt, E. (2016). *Cassandra: The Definitive Guide: Distributed Data at Web Scale*. O'Reilly Media.
- Chodorow, K. (2013). *MongoDB: The Definitive Guide*. O'Reilly Media.
- Clayberg, E., & Rubel, D. (2008). *Eclipse Plug-ins (3rd ed.)*. Addison-Wesley Professional.
- Cleveland, W. S. (1993). *Visualizing Data*. Hobart Press.
- CodeLite. (2023). *CodeLite: A Free, Open Source, Cross Platform C, C++, PHP and JavaScript IDE*. Obtenido de <https://codelite.org/>
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6), 377-387.
- Crockford, D. (2008). *JavaScript: The Good Parts*. O'Reilly Media.
- Chrome Web Store. (2023). *Chrome Web Store*. Obtenido de <https://chrome.google.com/webstore/category/extensions>
- Cui, Y., Kara, S., & Chan, K. C. (2020). Manufacturing big data ecosystem: A systematic literature review. *Robotics and computer-integrated Manufacturing*, 62, 101861.
- Dabit, N. (2019). *React Native in Action: Developing iOS and Android Apps with JavaScript*. Manning Publications.
- DataStax. (2023). *Introduction to Apache Cassandra*. <https://www.datastax.com/cassandra>
- Dayley, B. (2014). *Node.js, MongoDB and AngularJS Web Development*. Addison-Wesley Professional.
- Demirkan, H., & Delen, D. (2013). Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud. *Decision Support Systems*, 55(1), 412-421.

- Desai, J. N., Pandian, S., & Vij, R. K. (2021). Big data analytics in upstream oil and gas industries for sustainable exploration and development: A review. *Environmental Technology & Innovation*, 21, 101186.
- Dewson, R. (2019). *Beginning SQL Server for Developers* (5th ed.). Apress.
- Dumbill, E. (2012). *Planning for Big Data*. O'Reilly Media, Inc.
- Eckel, B. (2006). *Thinking in Java* (4th ed.). Prentice Hall.
- Eclipse Foundation. (2023). Eclipse IDE. Obtenido de <https://www.eclipse.org/ide/>
- edraw. (s.f.). Wondershare. Obtenido de <https://www.edrawsoft.com/es/radar-chart/>
- Electron. (2023). Build cross-platform desktop apps with JavaScript, HTML, and CSS. Obtenido de <https://www.electronjs.org/>
- Eisenman, B. (2015). *Learning React Native: Building Native Mobile Apps with JavaScript*. O'Reilly Media.
- Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems* (7th ed.). Pearson.
- El Mejor Software. (s.f.). Obtenido de <https://elmejorsoftware.com/mejores-software-de-programacion/>
- Embarcadero Technologies. (2023). Delphi: Develop cross-platform native apps fast. Obtenido de <https://www.embarcadero.com/products/delphi>
- Eran, E. (2019). CodeLite: An Open Source IDE for C/C++ Development. In *Open Source IDEs for Enterprise Application Development* (pp. 123-145). Packt Publishing.
- Etecé, E. (13 de agosto de 2024). Concepto. Obtenido de <https://concepto.de/histograma/>
- Everitt, B. S., & Skrondal, A. (2010). *The Cambridge Dictionary of Statistics*. Cambridge University Press.

- Facebook. (2023). React Native: A framework for building native apps using React. Obtenido de <https://reactnative.dev/>
- Facebook Inc. (2023). React – A JavaScript library for building user interfaces. <https://reactjs.org/>
- Few, S. (2009). *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press.
- Few, S. (2007). Save the Pies for Dessert. *Visual Business Intelligence Newsletter*, 1-14.
- Flanagan, D. (2020). *JavaScript: The Definitive Guide (7th ed.)*. O'Reilly Media.
- Flight, T. (2018). *Atom: A hackable text editor for the 21st century*. Packt Publishing.
- Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional.
- Fowler, M. (2018). *Refactoring: Improving the Design of Existing Code (2nd ed.)*. Addison-Wesley Professional.
- Friendly, M. (2009). Milestones in the history of thematic cartography, statistical graphics, and data visualization. In *Handbook of Data Visualization* (pp. 15-56). Springer.
- Friendly, M., & Denis, D. (2005). The early origins and development of the scatterplot. *Journal of the History of the Behavioral Sciences*, 41(2), 103-130.
- Gallardo, D., Burnette, E., & McGovern, R. (2012). *Eclipse in Action: A Guide for Java Developers*. Manning Publications.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). *Database systems: The complete book (2nd ed.)*. Pearson.

- Garsiel, T., & Irish, P. (2011). How Browsers Work: Behind the scenes of modern web browsers. HTML5 Rocks. Obtenido de <https://www.html5rocks.com/en/tutorials/internals/howbrowserswork/>
- GitHub. (2023). Atom: A hackable text editor for the 21st Century. Obtenido de <https://github.com/atom/atom>
- GitHub. (2022). Teletype for Atom. Obtenido de <https://teletype.atom.io/>
- Goodman, D. (2010). JavaScript & DHTML Cookbook (3rd ed.). O'Reilly Media.
- Google. (2023). Chrome Browser. Obtenido de <https://www.google.com/chrome/>
- Google. (2023c). Chrome Security. Obtenido de <https://www.google.com/chrome/security/>
- Google. (2023). Chrome V8. Obtenido de <https://v8.dev/>
- Google Developers. (2023). Chrome DevTools. <https://developers.google.com/web/tools/chrome-devtools>
- Google. (2023). Flutter - Beautiful native apps in record time. Obtenido de <https://flutter.dev/>.
- Google. (2023). Dart programming language. Obtenido de <https://dart.dev/>.
- Gosling, J., Joy, B., Steele, G., & Bracha, G. (2005). The Java Language Specification (3rd ed.). Addison-Wesley Professional.
- Hassani, H., Unger, S., & Beneki, C. (2020). Big data and actuarial science. *Big Data and Cognitive Computing*, 4(4), 40.
- Haverbeke, M. (2018). *Eloquent JavaScript: A Modern Introduction to Programming* (3rd ed.). No Starch Press.
- Heffelfinger, D. R. (2015). *Java EE 7 Development with NetBeans 8*. Packt Publishing.
- Hennessy, D. (2016). *Building Tools with GitHub: Customize Your Workflow*. O'Reilly Media.
- Hewitt, E. (2010). *Cassandra: The Definitive Guide*. O'Reilly Media.

- Holmes, J. (2019). *Ionic 4: Build iOS, Android, and Web Apps with Ionic & Angular*. Leanpub.
- Holzner, S. (2004). *Eclipse: A Java Developer's Guide*. O'Reilly Media.
- Horstmann, C. S. (2013). *Core Java Volume I--Fundamentals (9th ed.)*. Prentice Hall.
- Hoque, S. (2020). *Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js*. Packt Publishing Ltd.
- Hows, D., Membrey, P., & Plugge, E. (2015). *MongoDB Basics*. Apress.
- Hunt, A., & Thomas, D. (1999). *The Pragmatic Programmer: From Journeyman to Master*. Addison-Wesley Professional.
- IBM. (2023). *IBM COBOL*. Obtenido de <https://www.ibm.com/products/cobol>
- IEA. (2021). *World Energy Outlook 2021*. International Energy Agency.
- Ikegwu, A. C., Nweke, H. F., Anikwe, C. V., Alo, U. R., & Okonkwo, O. R. (2022). Big data analytics for data-driven industry: a review of data sources, tools, challenges, solutions, and research directions. *Cluster Computing*, 25(5), 3343-3387.
- Ionic. (2023). *Ionic Framework: Cross-Platform Mobile App Development*. Obtenido de <https://ionicframework.com/>
- Iqbal, R., Doctor, F., More, B., Mahmud, S., & Yousuf, U. (2020). Big Data analytics and Computational Intelligence for Cyber-Physical Systems: Recent trends and state of the art applications. *Future Generation Computer Systems*, 105, 766-778.
- Janssen, T. (2019). *Mastering Pascal and Delphi Programming*. Packt Publishing.
- JMP Statistical Discovery. (s.f.). Obtenido de [https://www.jmp.com/es\\_co/statistics-knowledge-portal/exploratory-data-analysis/line-graph.html](https://www.jmp.com/es_co/statistics-knowledge-portal/exploratory-data-analysis/line-graph.html)
- Johnson, B. (2022). *Visual Studio Code Distilled: A Comprehensive Guide to Enhancing Your Development Workflow*. Apress.

- Johnson, M. (2020). *Web Design with HTML, CSS, JavaScript and jQuery Set*. Wiley.
- Johnson, R. (2004). *Expert One-on-One J2EE Development without EJB*. Wiley Publishing, Inc.
- Josuttis, N. M. (2012). *The C++ Standard Library: A Tutorial and Reference (2nd ed.)*. Addison-Wesley Professional.
- Kahn, M. (2015). *Delphi Programming for Beginners*. CreateSpace Independent Publishing Platform.
- KEEP CODING Tech School. (s.f.). Obtenido de <https://keepcoding.io/blog/5-lenguajes-de-programacion-mas-usados-2022/#Python>
- Kiely, T. (2020). *Learn Next.js: Build and Deploy Production Ready React Applications*. Packt Publishing.
- Kirk, A. (2016). *Data Visualisation: A Handbook for Data Driven Design*. SAGE Publications.
- Knaflic, C. N. (2015). *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Wiley.
- Kreibich, J. A. (2010). *Using SQLite*. O'Reilly Media, Inc.
- Lakshman, A., & Malik, P. (2010). Cassandra: a decentralized structured storage system. *ACM SIGOPS Operating Systems Review*, 44(2), 35-40.
- Lazarus Team. (2023). *Lazarus Documentation*. Obtenido de <https://www.lazarus-ide.org/>
- Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012). Software development life cycle AGILE vs traditional approaches. *International Conference on Information and Network Technology*, 37(1), 162-167.
- Leffingwell, D., & Widrig, D. (2003). *Managing Software Requirements: A Use Case Approach (2nd ed.)*. Addison-Wesley Professional.

- Lippman, S. B., Lajoie, J., & Moo, B. E. (2012). C++ Primer (5th ed.). Addison-Wesley Professional.
- Lutz, M. (2013). Learning Python (5th ed.). O'Reilly Media.
- Lynch, J. (2021). Next.js: The Ultimate Guide to Building Server-rendered React Applications. O'Reilly Media.
- MariaDB Corporation. (2022). About MariaDB. Obtenido de <https://mariadb.com/about/>
- MariaDB Foundation. (2023). About MariaDB. Obtenido de <https://mariadb.org/about/>
- Marinakos, V., Doukas, H., Tsapelas, J., Mouzakitis, S., Sicilia, Á., Madrazo, L., & Sgouridis, S. (2020). From big data to smart energy services: An application for intelligent energy management. *Future Generation Computer Systems*, 110, 572-586.
- Marques, M. (2011). Bases de datos. Universitat Jaume I.
- Martin, R. C. (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.
- Martinez Unanue, R., Gortazar Bellas, F., & Fresno Fernandez, V. (2016). Lenguajes de programación y procesadores. Centro de Estudios Ramon Areces SA.
- McConnell, S. (2004). Code Complete: A Practical Handbook of Software Construction (2nd ed.). Microsoft Press.
- McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media.
- Meier, R. (2010). Professional Android 2 Application Development. Wiley Publishing.
- Membrey, P., Plugge, E., & Hawkins, T. (2010). The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing. Apress.
- Meyers, S. (2014). Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14. O'Reilly Media.

- Microsoft. (2023). C# documentation. Obtenido de <https://docs.microsoft.com/en-us/dotnet/csharp/>
- Microsoft. (2023). SQL Server Analysis Services. Obtenido de <https://docs.microsoft.com/en-us/analysis-services/>
- Microsoft. (2023). SQL Server 2022. Obtenido de <https://www.microsoft.com/en-us/sql-server/sql-server-2022>
- Microsoft. (2023). Visual Studio documentation. <https://docs.microsoft.com/en-us/visualstudio/>
- Microsoft. (2023). Visual Studio Code - Code Editing. Redefined. Obtenido de <https://code.visualstudio.com/>
- Mohammadpoor, M., & Torabi, F. (2020). Big Data analytics in oil and gas industry: An emerging trend. *Petroleum*, 6(4), 321-328.
- Momjian, B. (2001). *PostgreSQL: Introduction and Concepts*. Addison-Wesley.
- MongoDB. (2023). MongoDB Documentation. Obtenido de <https://www.mongodb.com/docs/>
- MongoDB, Inc. (2023). What Is MongoDB? – Obtenido de <https://www.mongodb.com/what-is-mongodb>
- Munawar, H. S., Ullah, F., Qayyum, S., & Shahzad, D. (2022). Big data in construction: current applications and future opportunities. *Big Data and Cognitive Computing*, 6(1), 18.
- Munzner, T. (2014). *Visualization Analysis and Design*. CRC Press.
- Murach, M. (2013). *Murach's Mainframe COBOL*. Mike Murach & Associates.
- Murray, S. (2017). *Interactive Data Visualization for the Web: An Introduction to Designing with D3*. O'Reilly Media.
- Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing (3rd ed.)*. John Wiley & Sons.

- MySQL. (n.d.). MySQL: The world's most popular open-source database. Obtenido de <https://www.mysql.com/>
- Napoli, A. (2020). Flutter Complete Reference: Create beautiful, fast and native apps for any device. Independently published.
- Nielsen, J. (2000). Designing Web Usability: The Practice of Simplicity. New Riders Publishing.
- Novick, V. (2017). React Native - Building Mobile Apps with JavaScript. Packt Publishing.
- Oliphant, T. E. (2007). Python for Scientific Computing. Computing in Science & Engineering, 9(3), 10-20.
- Oracle. (2023). Java Technology. Retrieved from <https://www.oracle.com/java/>
- Oracle. (2023). MySQL: Overview. Obtenido de <https://www.oracle.com/mysql/>
- Oracle. (2023). Oracle Database. Obtenido de <https://www.oracle.com/database/>
- Osmani, A. (2014). Learning JavaScript Design Patterns. O'Reilly Media.
- Owens, M. (2006). The Definitive Guide to SQLite. Apress.
- Papp, A. (2020). The Next.js Handbook. Flavio Copes.
- Patel, S. K. (2021). Pro SQL Server 2019 Administration: A Guide for the Modern DBA. Apress.
- Patton, R. (2006). Software Testing (2nd ed.). Sams Publishing.
- Pecanac, V. (2021). Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers. Packt Publishing.
- Perrons, R. K., & Jensen, J. W. (2015). Data as an asset: What the oil and gas sector can learn from other industries about "Big Data". Energy Policy, 81, 117-121.
- Petri, J. (2013). NetBeans Platform 6.9 Developer's Guide. Packt Publishing.
- Plugge, E., Membrey, P., & Hawkins, T. (2015). The Definitive Guide to MongoDB: A complete guide to dealing with Big Data using MongoDB. Apress.

- PostgreSQL Global Development Group. (n.d.). About PostgreSQL. Obtenido de <https://www.postgresql.org/about/>
- Prata, S. (2012). C++ Primer Plus (6th ed.). Addison-Wesley Professional.
- Pressman, R. S., & Maxim, B. R. (2015). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education.
- Price, M. J. (2020). C# 9 and .NET 5 – Modern Cross-Platform Development: Build intelligent apps, websites, and services with .NET 5, ASP.NET Core 5, and EF Core 5 using Visual Studio Code. Packt Publishing.
- Prieto, G. T. (2016). Platzi. Obtenido de [https://platzi.com/tutoriales/1050-programacion-basica/174-los-10-lenguajes-de-programacion-mas-usados-de-la-actualidad-2/?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=17739691128&utm\\_adgroup=&utm\\_content=&gclid=Cj0KCQiAkMGcBhCSARIsAIW6d0BGMTGq1aJwDXDfvi](https://platzi.com/tutoriales/1050-programacion-basica/174-los-10-lenguajes-de-programacion-mas-usados-de-la-actualidad-2/?utm_source=google&utm_medium=cpc&utm_campaign=17739691128&utm_adgroup=&utm_content=&gclid=Cj0KCQiAkMGcBhCSARIsAIW6d0BGMTGq1aJwDXDfvi)
- Python Software Foundation. (2023). Python Documentation. Obtenido de <https://docs.python.org/3/>
- Raschka, S. (2015). Python Machine Learning. Packt Publishing Ltd.
- Razzoli, F. (2014). Mastering MariaDB. Packt Publishing Ltd.
- Resig, J., & Bibeault, B. (2013). Secrets of the JavaScript Ninja. Manning Publications.
- Ricardo, C. M. (2009). Bases de datos. McGraw Hill Education.
- Roberts, T. (2020). Mastering Next.js: Build Powerful Server-side Applications with React. Independently Published.
- Sagiroglu, S., & Sinanc, D. (2013). Big data: A review. 2013 International Conference on Collaboration Technologies and Systems (CTS), 42-47.

- Sahni, V., Chopde, A., Goswami, M., & Kumar, A. (2024). Mern (Mongodb, Express-Js, React-Js, Node-Js) Stack Web-Based Themefied Education Platform For Placement Preparation. *Educational Administration: Theory and Practice*, 30(5), 1918-1928.
- Sande, W., & Galloway, E. (2021). *Flutter Apprentice: Learn to Build Cross-Platform Apps*. Razeware LLC.
- Salter, D., & Dantas, R. (2014). *NetBeans IDE 8 Cookbook*. Packt Publishing.
- Schildt, H. (2019). *Java: The Complete Reference (11th ed.)*. McGraw-Hill Education.
- Schwartz, B., Zaitsev, P., & Tkachenko, V. (2012). *High Performance MySQL: Optimization, Backups, and Replication*. O'Reilly Media, Inc.
- Shiffman, D. (2024). *The Nature of Code: Simulating Natural Systems with JavaScript*. San Francisco: No Starch Press.
- Sierra, K., & Bates, B. (2015). *Head First Java (2nd ed.)*. O'Reilly Media.
- Simpson, K. (2015). *You Don't Know JS: Up & Going*. O'Reilly Media.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database system concepts (7th ed.)*. McGraw-Hill Education.
- Skeet, J. (2019). *C# in Depth (4th ed.)*. Manning Publications.
- Smil, V. (2017). *Energy and Civilization: A History*. MIT Press.
- Smith, J. (2018). *Cross-Platform Development with Lazarus and Free Pascal*. Apress.
- Smith, J. (2019). *Mastering Adobe Dreamweaver CC 2019*. Packt Publishing.
- Smith, J., & Williams, R. (2022). *Enterprise Database Management: Strategies and Solutions*. McGraw-Hill.
- Smyth, N. (2020). *Visual Studio Code: Efficient Editing for Software Developers*. Payload Media.

- Soloaga, C. D. (6 de octubre de 2020). Socialmediapymes. Obtenido de <https://www.socialmediapymes.com/graficos-de-barras-cuando-y-como/>
- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.
- Sovacool, B. K., Heffron, R. J., McCauley, D., & Goldthau, A. (2016). Energy decisions reframed as justice and ethical concerns. *Nature Energy*, 1(5), 1-6.
- Spence, I. (2005). No Humble Pie: The Origins and Usage of a Statistical Chart. *Journal of Educational and Behavioral Statistics*, 30(4), 353-368.
- SQLite. (n.d.). About SQLite. Obtenido de <https://www.sqlite.org/about.html>
- StatCounter. (2023). Browser Market Share Worldwide. Obtenido de <https://gs.statcounter.com/browser-market-share>
- Stern, N., & Stern, R. A. (2014). *Structured COBOL Programming*. Wiley.
- Stonebraker, M., & Kemnitz, G. (1991). The POSTGRES next-generation database management system. *Communications of the ACM*, 34(10), 78-92.
- Stonebraker, M. (2018). The future of data management systems. *Communications of the ACM*, 61(9), 72-80.
- Stroustrup, B. (2013). *The C++ Programming Language* (4th ed.). Addison-Wesley Professional.
- Sullivan, D. (2015). *NoSQL for Mere Mortals*. Addison-Wesley Professional.
- Summerfield, M. (2010). *Programming in Python 3: A Complete Introduction to the Python Language* (2nd ed.). Addison-Wesley Professional.
- Support Google. (s.f.). Google. Obtenido de <https://support.google.com/docs/answer/9142677?hl=es-419>
- Tanenbaum, A. S., & Van Steen, M. (2007). *Distributed Systems: Principles and Paradigms* (2nd ed.). Prentice Hall.

- Teti, D. (2019). Hands-On Design Patterns with Delphi. Packt Publishing.
- Trejos Buritica, O. I. (2022). Logica de programación. Ediciones de la U.
- Troelsen, A., & Japikse, P. (2021). Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming. Apress.
- TuDashboard. (16 de Julio de 2020). Gráfica de Pastel. Conoce sus características y cuándo utilizarla. Obtenido de <https://tudashboard.com/grafica-de-pastel/>
- TuDashboard. (18 de agosto de 2021). Gráfico de burbuja: Características y usos más comunes. Obtenido de <https://tudashboard.com/grafico-de-burbuja/>
- Tufte, E. R. (2001). The Visual Display of Quantitative Information (2nd ed.). Graphics Press.
- Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. CreateSpace Independent Publishing Platform.
- Vercel. (2023). Next.js: The React Framework for Production. <https://nextjs.org/>
- Villalba, C. M., Uruquia Moraleda, A., & Rubio Gonzales, M. A. (2021). Lenguajes de programacion. UNED.
- Vogel, L. (2015). Eclipse IDE Tutorial. vogella GmbH. Obtenido de <https://www.vogella.com/tutorials/Eclipse/article.html>
- Wagner, B. (2021). Effective C#: 50 Specific Ways to Improve Your C# (3rd ed.). Addison-Wesley Professional.
- Ware, C. (2012). Information Visualization: Perception for Design (3rd ed.). Morgan Kaufmann.
- Wexler, J. (2020). Get Programming with Node.js. Manning Publications.
- Wexler, J. (2021). Next.js in Action. Manning Publications.
- Wielenga, G. (2015). Beginning NetBeans IDE: For Java Developers. Apress.
- Williams, R. (2018). The Non-Designer's Design Book. Peachpit Press.

Windmill, E. (2021). Flutter in Action. Manning Publications.

Windmill, E. (2020). Flutter in Action. Simon and Schuster.

Wixom, B. H., & Gao, J. (2015). Mobile Application Development with React Native. Communications of the Association for Information Systems, 37, 1.

Wu, W. (01 de marzo de 2018). React Native vs Flutter, cross-platform mobile application frameworks.

Yergin, D. (2020). The New Map: Energy, Climate, and the Clash of Nations. Penguin Press.

Zakas, N. C. (2012). Professional JavaScript for Web Developers (3rd ed.). Wrox.

Zelle, J. (2016). Python Programming: An Introduction to Computer Science (3rd ed.). Franklin, Beedle & Associates Inc.