UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

REPORTE DE INVESTIGACIÓN DE LOS LENGUAJES Y BIBLIOTECAS PARA ANÁLISIS Y PROCESAMIENTO DE DATOS

MATERIA: Extracción de Conocimiento en Bases de Datos

MAESTR@: Enrique Mascote

ALUMNO: Carlos Adrián Mata Nevárez

GRUPO: IDGS91N

FECHA:23/09/2025

ÍNDICE

INTRODUCCIÓN

Hoy en día los datos tienen un papel muy importante en empresas, escuelas y en la vida diaria. Con ellos se pueden tomar decisiones más claras, descubrir patrones y encontrar maneras de mejorar distintas actividades. Para trabajar con datos de manera más fácil y rápida, se usan lenguajes de programación y bibliotecas que ayudan a organizarlos, revisarlos y sacar conclusiones.

Conocer los lenguajes más usados es necesario porque cada uno sirve mejor en diferentes situaciones. Python, por ejemplo, se usa mucho porque es sencillo y tiene muchas bibliotecas disponibles. R se usa más en estudios e investigaciones, sobre todo cuando se necesita analizar encuestas o estadísticas. SQL es la base para consultar bases de datos grandes, mientras que Scala y Julia sirven cuando se trabaja con grandes cantidades de información o cálculos más pesados.

El objetivo de este reporte es mostrar una idea general de estos lenguajes, junto con ejemplos básicos que enseñan cómo cargar un archivo de datos y ver un resumen inicial. Esto permite entender mejor cómo se aplican y en qué casos funcionan de manera más útil.

LENGUAJES

Python

Descripción general:
Phython es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo (aws amazon, 2025).

Paradigma: Interpretado, tipado dinámico, multiparadigma (procedural, orientado a objetos, funcional limitado).

Ámbito de uso principal: Ciencia de datos, ML/IA, desarrollo backend, scripting, automatización.

Bibliotecas y frameworks clave
Pandas: Manipulación de datos en estructuras DataFrame; lectura/escritura CSV, limpieza, filtrado y agregaciones.

Caso de uso: Preprocesamiento de datos tabulares antes de entrenar un modelo.

NumPy: Operaciones numéricas con arrays multidimensionales; base para muchas bibliotecas científicas.

Caso de uso: Cálculos vectorizados y álgebra lineal en modelos ML.

scikit-learn: Modelos clásicos de ML (regresión, clasificación, clustering), pipelines y evaluación.

Caso de uso: Prototipado rápido de modelos supervisados (Diaz, 2023).

Caso de uso o ejemplo concreto

Un caso en el que yo usaría Python sería para analizar datos de una empresa que registra ventas en CSV. Con pandas podría cargar y limpiar la información, y después con scikit-learn entrenar un modelo sencillo que prediga las ventas futuras. Esto sería útil en un negocio que quiere anticipar la demanda de productos y organizar mejor su inventario.

Mini-ejemplo

```python
import pandas as pd


# Cargar CSV ficticio
df = pd.read_csv('clientes.csv')


# Mostrar primeras filas
print(df.head())


# Resumen básico
print(df.describe())
```

Python.py | clientes.csv

clientes.csv

```
Nombre,Edad,Ciudad,Compras
Ana,28,CDMX,5
Luis,34,Yucatan,3
Carlos,22,Guadalajara,8
Marta,45,CDMX,2
Jorge,38,Monterrey,6
```

```
  import pandas as pd
   Nombre  Edad       Ciudad  Compras
0     Ana    28         CDMX        5
1    Luis    34      Yucatan        3
2  Carlos    22  Guadalajara        8
3   Marta    45         CDMX        2
4   Jorge    38    Monterrey        6
            Edad   Compras
count   5.000000  5.000000
mean   33.400000  4.800000
std     8.876936  2.387467
min    22.000000  2.000000
25%    28.000000  3.000000
50%    34.000000  5.000000
75%    38.000000  6.000000
max    45.000000  8.000000
```

R

Descripción general:

R es un entorno de software libre (licencia GNU GLP) y lenguaje de programación interpretado, es decir, ejecuta las instrucciones directamente, sin una previa compilación del programa a instrucciones en lenguaje máquina. El término entorno, en R, se refiere a un sistema totalmente planificado y coherente, en lugar de una acumulación de herramientas específicas e inflexibles, como suele ser el caso en otros softwares de análisis de datos (unir, 2023)

Paradigma: Interpretado, tipado dinámico, orientado a objetos (S3/S4), funcional.

Ámbito de uso principal: Análisis estadístico, visualización, investigación académica y modelado estadístico.

Bibliotecas y frameworks clave

tidyverse (ggplot2, dplyr, tidyr) — Conjunto de paquetes para manipulación y visualización de datos con una gramática coherente.

Caso de uso: Limpieza y transformación de datos y creación de gráficos avanzados.

data.table — Manipulación de datos altamente eficiente y optimizada en memoria.
Caso de uso: Procesamiento rápido de tablas grandes en análisis exploratorio.

caret / tidymodels — Frameworks para entrenamiento, validación y selección de modelos.

Caso de uso: Pipeline de modelado y evaluación cruzada.

Caso de uso o ejemplo concreto

R lo imagino más en un contexto académico o de investigación, por ejemplo, cuando se tiene una encuesta con cientos de respuestas. Con R se pueden aplicar pruebas estadísticas y usar ggplot2 para hacer gráficas claras que ayuden a interpretar los resultados. Un caso sería comparar la relación entre horas de estudio y calificaciones en estudiantes de una universidad.

Mini-ejemplo

Scala

Descripción general:

Scala es un lenguaje puramente orientado a objetos en el sentido de que todo es un objeto. Los tipos y comportamientos de objetos son descritos por clases y traits (que podría ser traducido como un “rasgo”). Las clases pueden ser extendidas a través de subclases y un mecanismo flexible de composición mezclada que provee un claro remplazo a la herencia múltiple (scala, 2025).

Paradigma: Compilado (JVM), tipado estático (con inferencia), multiparadigma (funcional y OO).

Ámbito de uso principal: Big Data (con Apache Spark), sistemas distribuídos y aplicaciones de alto rendimiento sobre JVM.

Bibliotecas y frameworks clave

tidyverse (ggplot2, dplyr, tidyr) — Conjunto de paquetes para manipulación y visualización de datos con una gramática coherente.

Caso de uso: Limpieza y transformación de datos y creación de gráficos avanzados.

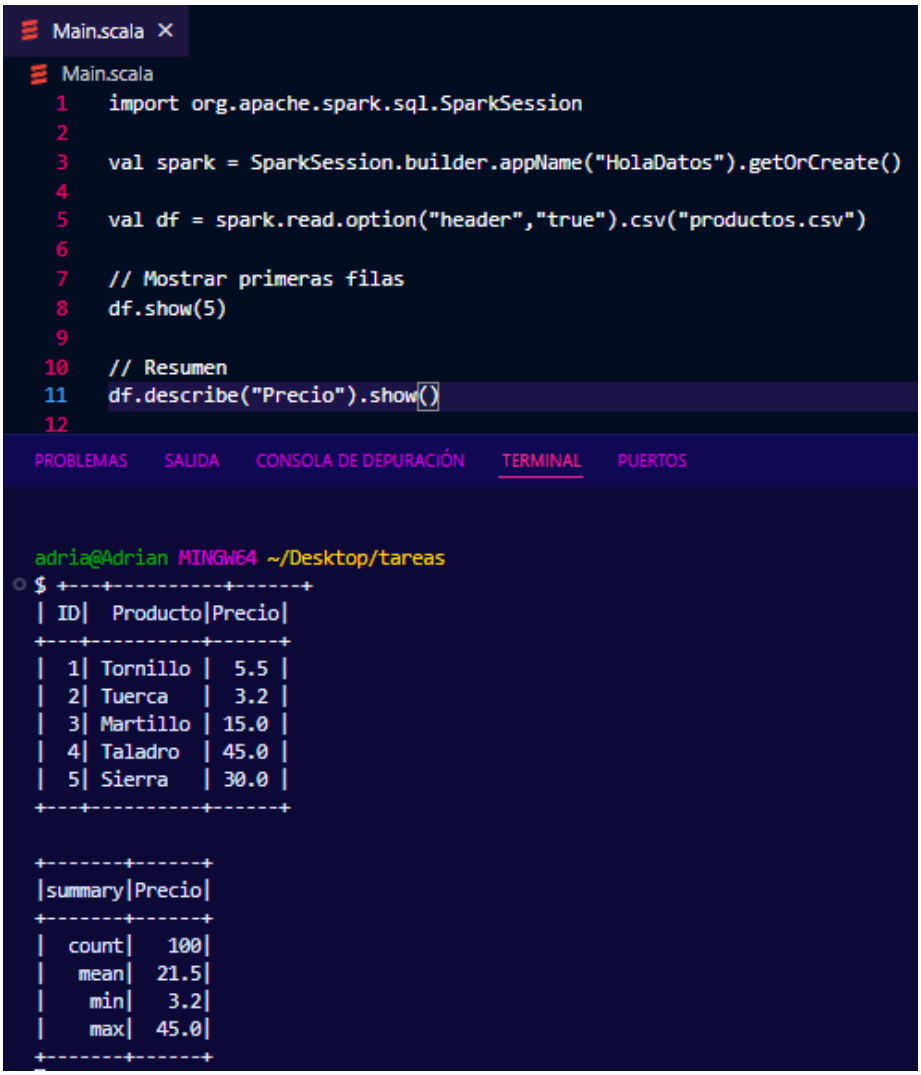
data.table — Manipulación de datos altamente eficiente y optimizada en memoria. Caso de uso: Procesamiento rápido de tablas grandes en análisis exploratorio.

caret / tidymodels — Frameworks para entrenamiento, validación y selección de modelos. Caso de uso: Pipeline de modelado y evaluación cruzada (Fernandez, 2024).

Caso de uso o ejemplo concreto

Scala es muy útil cuando hay que trabajar con datos masivos en tiempo real. Un ejemplo sería en una plataforma de streaming que necesita analizar qué canciones escuchan los usuarios al momento, para recomendar contenido personalizado. Gracias a Spark en Scala se puede procesar la información a gran velocidad y alimentar un sistema de recomendaciones en vivo.

Mini-ejemplo



SQL

Descripción general:

SQL es un lenguaje de computación para trabajar con conjuntos de datos y las relaciones entre ellos. Los programas de bases de datos relacionales, como Microsoft Office Access, usan SQL para trabajar con datos (microsoft, 2025).

Paradigma: Declarativo, específico para bases de datos.

Ámbito de uso principal: Consulta, transformación y extracción de datos almacenados en sistemas RDBMS (PostgreSQL, MySQL, SQL Server, Oracle).

Bibliotecas y frameworks clave

PostgreSQL / MySQL / SQL Server — Sistemas gestores de bases de datos con extensiones para análisis y funciones de ventana.

Caso de uso: Extracción de features y agregaciones para ML.

SQLAlchemy (Python) — ORM y capa de abstracción para ejecutar consultas SQL desde aplicaciones Python.

Caso de uso: Integración de pipelines de datos con aplicaciones.

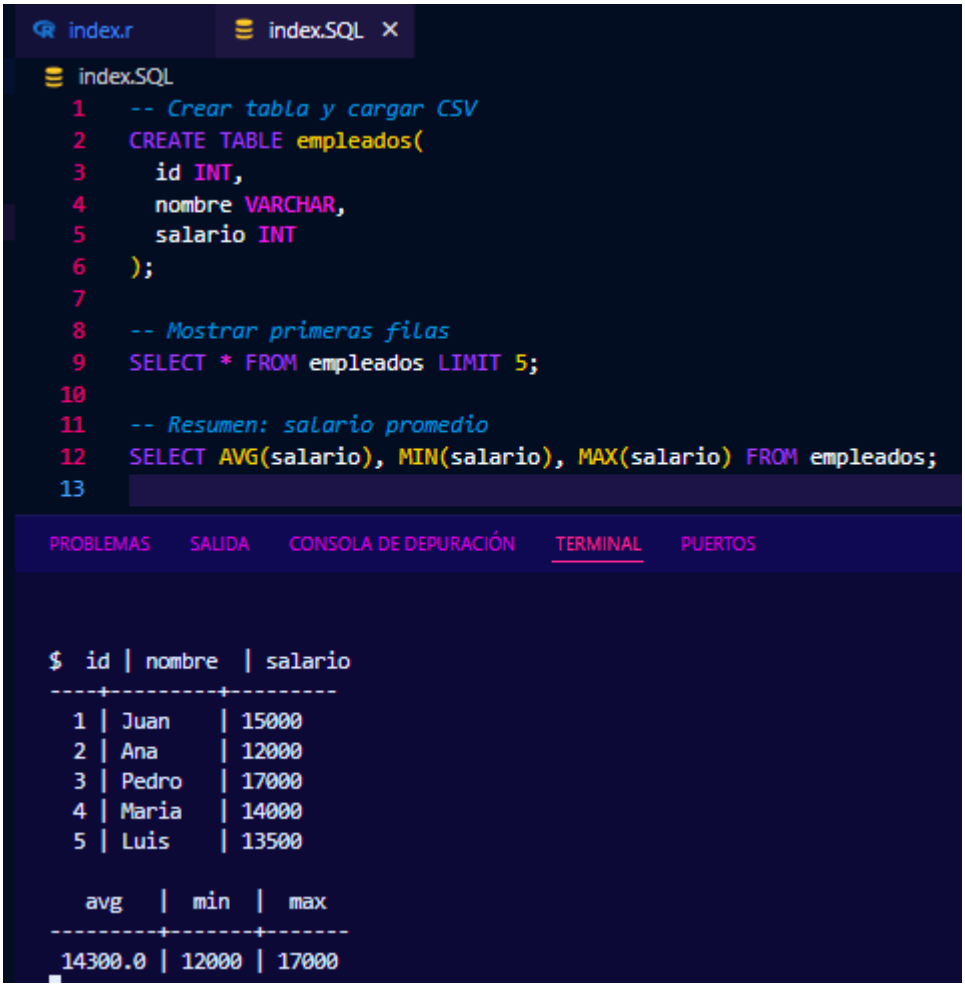
BigQuery / Snowflake — Data warehouses escalables con SQL para análisis a gran escala.

Caso de uso: Consultas analíticas en Big Data (Microsoft Ignite, 2025).

Caso de uso o ejemplo concreto

SQL es práctico en cualquier situación donde hay bases de datos grandes. Un caso concreto sería en una empresa de renta de autos: con SQL se puede consultar qué autos están disponibles, cuáles tienen más uso y hasta generar reportes de ganancias por mes. Es una forma rápida de obtener respuestas puntuales a preguntas de negocio directamente de la base de datos.

Mini-ejemplo



Julia

Descripción general:

Julia es un lenguaje de programación dinámico, compilado justo a tiempo (JIT), de alto nivel, diseñado para ofrecer a los usuarios la velocidad de C/C++ y al mismo tiempo ser tan fácil de usar como Python. Desarrollado en el MIT por el Dr. Viral Shah, el Prof. Alan Edelman, el Dr. Jeff Bezanson y Stefan Karpinski en 2012 (guides libraries, 2025).

Paradigma: Compilado (JIT), tipado dinámico con posibilidad de tipado estático, multiparadigma (procedural, funcional, orientado a objetos limitado).

Ámbito de uso principal: Análisis numérico, computación científica, prototipado de algoritmos numéricos y ML (Coursera, 2024).

Bibliotecas y frameworks clave

DataFrames.jl — Manipulación de tablas estilo DataFrame (similar a pandas).

Caso de uso: Limpieza y transformación de datos en Julia.

CSV.jl — Lectura/escritura eficiente de archivos CSV.

Caso de uso: Importar datasets para análisis numérico.

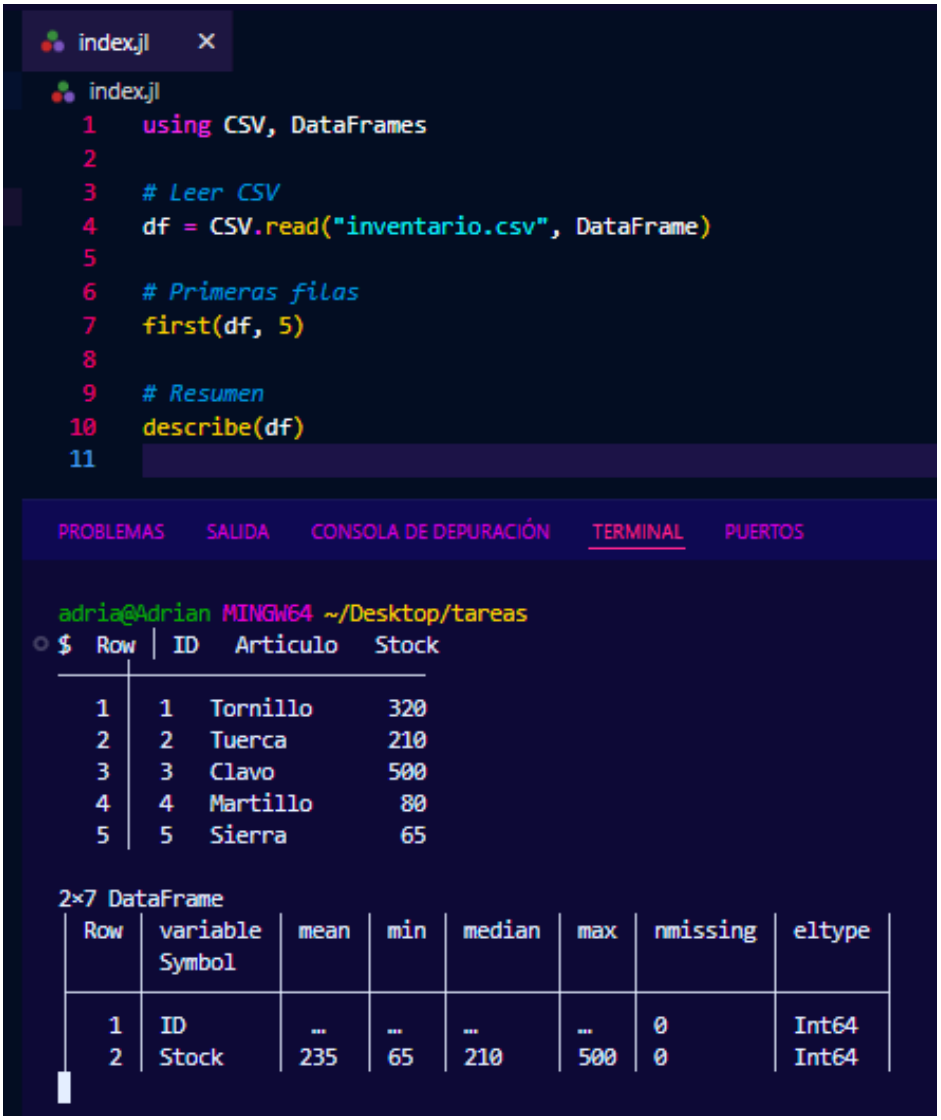
Flux.jl — Framework de deep learning en Julia.

Caso de uso: Prototipado de redes neuronales con buena performance.

Caso de uso o ejemplo concreto

Julia me parece más orientado a entornos científicos o de investigación avanzada, donde se necesita mucha rapidez en cálculos matemáticos. Un caso concreto sería en el área de ingeniería, por ejemplo, para simular el comportamiento de un material bajo diferentes condiciones de presión y temperatura. Julia ayuda porque combina la facilidad de un lenguaje moderno con un rendimiento cercano al de C.

Mini-ejemplo



CONCLUSIÓN

Al comparar los distintos lenguajes investigados, se notan tanto puntos en común como diferencias importantes. Todos permiten trabajar con datos, cargarlos y obtener resultados útiles, pero cada uno lo hace de forma distinta. Python y R se parecen en que ambos resultan fáciles de aprender y cuentan con bibliotecas muy completas. Sin embargo, Python es más general y puede usarse en muchas áreas, mientras que R se concentra más en lo estadístico y en la creación de gráficos.

SQL es diferente porque no se utiliza para programar modelos o cálculos complejos, sino para consultar y organizar datos en bases estructuradas. Aun así, es básico porque en muchos proyectos los datos siempre terminan guardados en una base de este tipo. Por otro lado, Scala y Julia comparten el hecho de que ofrecen más velocidad y potencia, aunque requieren más experiencia. Scala se usa más en contextos de big data y Julia en cálculos científicos o matemáticos avanzados.

Tomando en cuenta el rendimiento, Python y R funcionan bien en proyectos pequeños o medianos, pero pueden volverse lentos en procesos demasiado grandes. SQL mantiene un buen desempeño en consultas directas, pero no cubre todas las necesidades de análisis. Scala y Julia, en cambio, destacan porque permiten trabajar con más datos o con operaciones más pesadas sin perder rapidez.

El ecosistema de bibliotecas también marca una diferencia. Python es el más amplio y variado, mientras que R mantiene un lugar sólido en lo estadístico. SQL se centra en su propio lenguaje de consultas. Scala aprovecha bibliotecas de big data como Spark, y Julia todavía tiene menos librerías que otros, pero crece cada vez más en entornos de investigación.

Así que, para proyectos de análisis ligero o de aprendizaje conviene usar Python o R, ya que resultan más accesibles y prácticos. Por otro lado, para proyectos de producción a gran escala, donde se manejan millones de registros o se necesitan cálculos avanzados, es mejor elegir Scala o Julia. SQL, aunque no se compara directamente, sigue siendo necesario como complemento en la mayoría de los escenarios.

REFERENCIAS

(23 de Mayo de 2023). Obtenido de unir: https://mexico.unir.net/noticias/ingenieria/lenguaje-r-big-data

aws amazon. (22 de Sep de 2025). Obtenido de https://aws.amazon.com/es/what-is/python

Coursera, p. d. (27 de Sep de 2024). *Coursera*. Obtenido de https://www-coursera-org.translate.goog/articles/julia-programming-language?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc

Diaz, D. (13 de Oct de 2023). Obtenido de kinsta: https://kinsta.com/es/blog/python-frameworks/#tipos-de-frameworks-de-python

Fernandez, O. (14 de Julio de 2024). *aprenderbigdata*. Obtenido de https://aprenderbigdata.com/introduccion-scala

guides libraries. (23 de Sep de 2025). Obtenido de https://guides-libraries-uc-edu.translate.goog/julia?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=wa

microsoft. (23 de Sep de 2025). Obtenido de https://support.microsoft.com/es-es/topic/access-sql-conceptos-básicos-vocabulario-y-sintaxis-444d0303-cde1-424e-9a74-e8dc3e460671#:~:text=SQL%20es%20un%20lenguaje%20de%20computación%20para%20trabajar%20con%20conjuntos,SQL%20para%20trabajar%20con%20datos.

Microsoft Ignite. (23 de Sep de 2025). Obtenido de https://learn.microsoft.com/es-es/sql/relational-databases/clr-integration/database-objects/supported-net-framework-libraries?view=sql-server-ver17

scala. (23 de Sep de 2025). Obtenido de https://docs.scala-lang.org/es/tour/tour-of-scala.html