UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

TECNOLOGÍAS DE LA INFORMACIÓN

EXTRACCIÓN DE CONOCIMIENTO EN BASES DE DATOS

I.4. REPORTE DE INVESTIGACIÓN DE LOS LENGUAJES Y BIBLIOTECAS PARA ANÁLISIS Y PROCESAMIENTO DE DATOS.

IDGS91N

PRESENTA:

SEBASTIÁN ACOSTA ORTIZ

DOCENTE:

LUIS ENRIQUE MASCOTE CANO

Chihuahua, Chih., 23 de septiembre de 2025

Índice

Introducción

Con un tema tan amplio como lo es el análisis de datos, considero crucial conocer los lenguajes y bibliotecas utilizados, ya que estos no solo facilitan la tarea, sino que también permiten resolver problemas de manera eficiente y automatizada. Entre las numerosas opciones disponibles, Python y SQL son esenciales por su facilidad de uso, versatilidad y gran demanda en el mercado laboral. Dominar estos lenguajes me permitirá crear pipelines de datos, realizar análisis estadísticos y diseñar visualizaciones, todo sin necesidad de aprender herramientas más complejas desde el principio.

Sumado a lo anterior, no me enfocaré únicamente en estos lenguajes; a lo largo del reporte abordaré otros bastante populares en su ámbito, tales como R, Scala, Julia y Java. Recalcaré sus ventajas, desventajas e importancia dentro de la ciencia de datos y sus múltiples campos, ya que hacen posible analizar grandes volúmenes de información, encontrar patrones y relaciones, predecir comportamientos y tomar decisiones basadas en datos.

En fin, el objetivo de este reporte es investigar los lenguajes de programación más utilizados en proyectos de IA, ML, DM y Big Data, entendiendo cómo cada uno aporta al manejo y análisis de datos. Además, me interesa identificar las bibliotecas más relevantes de cada lenguaje para conocer las herramientas que facilitan el trabajo en ciencia de datos. También analizaré sus ventajas y limitaciones en cuanto a facilidad de uso, rendimiento y aplicaciones reales, así como incluir ejemplos sencillos que muestren su utilidad en tareas básicas de análisis. Finalmente, reflexionaré sobre cuál o cuáles considero más apropiados según el tipo de proyecto, ya sea para análisis ligeros o para implementaciones a gran escala.

Python

Python es un lenguaje interpretado, de tipado dinámico y multiparadigma (orientado a objetos, funcional y procedural). Su principal ventaja radica en su sintaxis clara y en un ecosistema de bibliotecas que lo han posicionado como el estándar en la ciencia de datos. Es el lenguaje más utilizado en machine learning, inteligencia artificial, minería de datos y Big Data, además de tener aplicaciones en el desarrollo web y la automatización.

Bibliotecas clave:

- Pandas: Manejo de datos tabulares con DataFrames. Ejemplo: limpieza y transformación de registros de ventas por región.
- Scikit-learn: Machine learning clásico (regresión, clasificación, clustering, reducción de dimensionalidad). Ejemplo: entrenar un modelo para predecir la deserción de clientes.
- TensorFlow (framework de deep learning): Redes neuronales y modelos de IA a gran escala. Ejemplo: reconocimiento de imágenes en sistemas de seguridad.

Mini-ejemplo "Hola datos":

```
import pandas as pd
df = pd.read_csv("datos.csv")
print(df.head())
```

R

R es un lenguaje interpretado y de tipado dinámico, creado específicamente para el análisis estadístico y la visualización de datos. Es el favorito de estadísticos, matemáticos y científicos sociales, ya que incluye de manera nativa muchas funciones de análisis avanzado. También destaca en entornos académicos e investigación, donde la visualización y la comunicación de resultados son cruciales.

Bibliotecas clave:

- ggplot2: Visualización avanzada con gramática de gráficos. Ejemplo: elaboración de gráficos de dispersión en un estudio de satisfacción del cliente.
- dplyr: Transformación de datos con funciones intuitivas para filtrar, agrupar y resumir. Ejemplo: análisis de encuestas en grandes poblaciones.
- caret: Marco integral para machine learning, que unifica entrenamiento y validación cruzada de modelos. Ejemplo: comparar algoritmos de clasificación en un mismo dataset.

Mini-ejemplo "Hola datos":

```
datos <- read.csv("datos.csv")
head(datos)
```

Scala

Scala es un lenguaje compilado, de tipado estático, que combina la programación orientada a objetos con la funcional. Su potencia y compatibilidad con Java lo convierten en una opción destacada en sistemas distribuidos, análisis de datos a gran escala y aplicaciones de alto rendimiento. Es especialmente importante en el ecosistema de Apache Spark, lo que lo posiciona como clave en Big Data.

Bibliotecas clave:

- Apache Spark (framework de Big Data): Procesamiento distribuido y machine learning con Spark MLlib. Ejemplo: analizar en tiempo real transacciones de una plataforma de e-commerce.
- Akka (framework de concurrencia): Creación de aplicaciones concurrentes y distribuidas. Ejemplo: sistemas de streaming en tiempo real.

- **Breeze**: Biblioteca matemática para álgebra lineal y cálculos numéricos. Ejemplo: simulaciones estadísticas complejas.

Mini-ejemplo "Hola datos":

```
import org.apache.spark.sql.SparkSession
val spark = SparkSession.builder.appName("HolaDatos").getOrCreate()
val df = spark.read.option("header", "true").csv("datos.csv")
df.show()
```

SQL

SQL (Structured Query Language) es un lenguaje declarativo y de tipado estático, utilizado exclusivamente para gestionar y consultar bases de datos relacionales. Es un estándar en la manipulación de datos, tanto en entornos empresariales como académicos. Aunque no es un lenguaje de propósito general, resulta indispensable para trabajar con grandes volúmenes de información estructurada y se integra con Python y R.

Bibliotecas clave (motores y extensiones):

- **PostgreSQL**: Motor de base de datos con funciones analíticas y extensiones geoespaciales (PostGIS). Ejemplo: análisis de rutas en aplicaciones de logística.
- **SQL Server**: Motor de Microsoft con integración en entornos corporativos y capacidades de BI. Ejemplo: reportes financieros automatizados en empresas.
- **BigQuery (servicio en la nube)**: Plataforma de Google Cloud para consultas masivas en petabytes de datos. Ejemplo: analizar registros de usuarios en tiempo real.

Mini-ejemplo "Hola datos":

```
SELECT * FROM datos
LIMIT 5;
```

Julia

Julia es un lenguaje compilado Just-In-Time (JIT), de tipado dinámico, que combina la facilidad de lenguajes como Python con el rendimiento de C. Está diseñado para cómputo científico, análisis numérico y aplicaciones que requieren alto rendimiento. A pesar de ser joven, ha crecido en popularidad dentro de comunidades de investigación, simulación y finanzas cuantitativas.

Bibliotecas clave:

- DataFrames.jl: Manipulación de datos tabulares similar a pandas en Python. Ejemplo: análisis de resultados de experimentos científicos.
- Flux.jl (framework de deep learning): Creación de modelos de redes neuronales. Ejemplo: procesamiento de lenguaje natural en Julia.
- Plots.jl: Visualización de datos flexible con integración en múltiples backends. Ejemplo: gráficas de series temporales.

Mini-ejemplo "Hola datos":

```
using DataFrames, CSV
df = CSV.read("datos.csv", DataFrame)
first(df, 5)
```

Java

Java es un lenguaje compilado, de tipado estático y orientado a objetos. Ha sido durante décadas un pilar en la industria del software, destacando por su estabilidad y escalabilidad. En el análisis de datos y la IA, Java es útil en Big Data, sistemas críticos y proyectos de producción a gran escala, aunque su sintaxis resulta más extensa en comparación con Python o R.

Bibliotecas clave:

- Weka: Biblioteca de minería de datos con algoritmos de clasificación, clustering y regresión. Ejemplo: segmentación de clientes en marketing.
- Deeplearning4j (framework de deep learning): Entrenamiento de redes neuronales con soporte para GPUs. Ejemplo: diagnóstico automático de imágenes médicas.
- Hadoop (ecosistema de Big Data): Procesamiento distribuido de datos masivos en clusters. Ejemplo: análisis batch de registros de servidores.

Mini-ejemplo "Hola datos":

```
import java.sql.*;
public class HolaDatos {
    public static void main(String[] args) throws Exception {
        Connection con =
DriverManager.getConnection("jdbc:sqlite:datos.db");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM datos LIMIT 5");
        while(rs.next()) {
            System.out.println(rs.getString(1));
        }
    }
}
```

Conclusión

Al comparar los lenguajes analizados, se puede apreciar que Python y R resultan útiles para análisis ligeros y exploratorios debido a la amplia cantidad de bibliotecas disponibles y su facilidad de uso. SQL continúa siendo la base para manipular datos en bases relacionales, manteniendo un papel indispensable en prácticamente cualquier proyecto. Por otro lado, Scala y Java destacan en escenarios de Big Data y producción a gran escala, donde el rendimiento

y la integración con sistemas distribuidos son prioritarios. Finalmente, Julia representa una alternativa interesante, ya que combina la sencillez de lenguajes modernos con un rendimiento muy alto, aunque su ecosistema aún está en desarrollo.

En síntesis, cada lenguaje tiene ventajas y limitaciones que lo hacen más apropiado según el tipo de proyecto: algunos se ajustan mejor a tareas de análisis exploratorio y visualización, mientras que otros son más adecuados para entornos de producción con grandes volúmenes de datos.

Referencias

Apache Spark. (2025). *MLlib: Machine Learning Library*. Apache Software Foundation. https://spark.apache.org/mllib/

DataFrames.jl. (2025). *Data manipulation in Julia*. JuliaData. https://dataframes.juliadata.org/

OpenAI. (2025). *ChatGPT (versión GPT-5) [Modelo de lenguaje de IA]*. Recuperado el 23 de septiembre de 2025 de https://chat.openai.com/

Pandas. (2025). *Python Data Analysis Library*. https://pandas.pydata.org/

Posit. (2025). *R packages for data science: dplyr, ggplot2, caret*. Posit Software. https://posit.co/

Scikit-learn. (2025). *Machine Learning in Python*. https://scikit-learn.org/

TensorFlow. (2025). *An end-to-end open source machine learning platform*. Google. https://www.tensorflow.org/

The University of Waikato. (2025). *Weka: Data Mining Software in Java*. Machine Learning Group. https://www.cs.waikato.ac.nz/ml/weka/