3. Desarrollo

3.1 Lenguajes para el desarrollo de software
3.2 Manejadores de bases de datos
3.3 Construcción del sistema

Los lenguajes de programación desempeñan un papel fundamental en la ingeniería de datos, ya que permiten la creación, gestión y análisis de grandes conjuntos de datos de manera eficiente.

Las necesidades del proyecto: El tipo de datos que se van a analizar, las herramientas y bibliotecas que se van a utilizar, y el entorno de desarrollo en el que se va a trabajar.

El nivel de experiencia del ingeniero de datos: Los lenguajes de programación como Python y R son buenas opciones para principiantes, mientras que lenguajes como Java y Scala pueden ser más desafiantes de aprender.

Las preferencias personales: Algunos ingenieros de datos prefieren un lenguaje de programación en particular por su sintaxis, su rendimiento o su comunidad de usuarios.

Nota: Los lenguajes de programación son una herramienta esencial para los ingenieros de datos. La elección del lenguaje de programación adecuado depende de una serie de factores, incluyendo las necesidades del proyecto, el nivel de experiencia del ingeniero de datos y las preferencias personales.

Python:

- Python es uno de los lenguajes de programación más populares en el campo de la ingeniería de datos debido a su simplicidad y versatilidad. Tiene una amplia variedad de bibliotecas, como Pandas, NumPy y SciPy, que facilitan la manipulación y el análisis de datos. También es compatible con frameworks de procesamiento de datos como Apache Spark y herramientas de aprendizaje automático como TensorFlow y Scikit-Learn.

- Puede ser más lento que algunos lenguajes compilados como C++ o Java en ciertas operaciones intensivas de CPU.

SQL (Structured Query Language):

- SQL es esencial para gestionar bases de datos relacionales y realizar consultas complejas. Es altamente eficiente en la recuperación y manipulación de datos en sistemas de gestión de bases de datos (DBMS) como MySQL, PostgreSQL y Microsoft SQL Server.

- No es un lenguaje de programación general, por lo que no es adecuado para tareas de procesamiento de datos más amplias, como el análisis avanzado o el aprendizaje automático.

R:

- R es un lenguaje de programación diseñado específicamente para la estadística y el análisis de datos. Tiene una gran cantidad de paquetes y librerías dedicados al análisis de datos y la visualización, lo que lo hace ideal para tareas de minería de datos y estadísticas.

- No es tan versátil como Python para aplicaciones más generales de desarrollo y no es tan eficiente en el procesamiento de grandes conjuntos de datos como otros lenguajes.

Java:

- Java es conocido por su rendimiento y escalabilidad. Es ampliamente utilizado en la construcción de sistemas de procesamiento de datos en tiempo real y aplicaciones de big data a través de frameworks como Apache Hadoop y Apache Flink.

- Puede tener una curva de aprendizaje más empinada en comparación con Python y R, y la escritura de código en Java tiende a ser más verbose.

Scala:

- Scala se ha vuelto popular en el contexto de Apache Spark debido a su concisión y la capacidad de aprovechar la programación funcional y orientada a objetos. Combina las ventajas de Java con la expresividad de lenguajes como Python o R.

- Requiere conocimientos de programación funcional que pueden ser un desafío para algunos desarrolladores.

Julia:

- Julia es un lenguaje de programación de alto rendimiento diseñado específicamente para la computación científica y el análisis de datos. Ofrece una velocidad comparable a la de lenguajes compilados como C++ y la facilidad de uso de Python o R.

- Aunque ha ganado tracción en la comunidad de la ciencia de datos, aún no es tan ampliamente adoptado como Python o R.

Spark (no es un lenguaje, sino un framework):

- Apache Spark se utiliza ampliamente en la ingeniería de datos para el procesamiento distribuido de grandes conjuntos de datos. Puede ser interactuado con lenguajes como Python, Scala y Java, lo que lo hace versátil y escalable.

- La configuración y administración de clústeres Spark pueden ser complicadas.

Nota:

La elección del lenguaje de programación para la ingeniería de datos depende de las necesidades específicas del proyecto, la familiaridad del equipo con el lenguaje y las herramientas disponibles. Python es una opción muy popular debido a su versatilidad y una amplia gama de bibliotecas, pero otros lenguajes como SQL, R, Java, Scala y Julia también tienen sus aplicaciones específicas en este campo. La elección final debe basarse en las características y requisitos del proyecto.

Referencias:

- Hennig, Christian & Hoppe, Tobias & Eisenmann, Harald & Viehl, Alexander & Bringmann, Oliver. (2016). SCDML: A Language for Conceptual Data Modeling in Model-based Systems Engineering. 184-192. 10.5220/0005676501840192.
- Furia, Carlo & Torkar, Richard & Feldt, Robert. (2022). Applying Bayesian Analysis Guidelines to Empirical Software Engineering Data: The Case of Programming Languages and Code Quality. ACM Transactions on Software Engineering and Methodology. 31. 1-38. 10.1145/3490953.
- Furia, Carlo & Torkar, Richard & Feldt, Robert. (2023). Towards Causal Analysis of Empirical Software Engineering Data: The Impact of Programming Languages on Coding Competitions. ACM Transactions on Software Engineering and Methodology. 10.1145/3611667.
- Thakre, Purushottam. (2023). Review on Software Technology. International Journal for Research in Applied Science and Engineering Technology. 11. 2865-2869. 10.22214/ijraset.2023.54096.
- Altherwi, Muna. (2019). An empirical study of programming language effect on open-source software development. 49-51. 10.1145/3359061.3361079.

3. Desarrollo

3.1 Lenguajes para el desarrollo de software
3.2 Manejadores de bases de datos
3.3 Construcción del sistema

Los manejadores de bases de datos son componentes fundamentales en la ingeniería de datos, ya que permiten almacenar, gestionar y consultar grandes volúmenes de información de manera eficiente. En este contexto, es esencial comprender cómo funcionan estos sistemas y cuáles son las opciones disponibles. A continuación, se presenta una investigación sobre el tema de los manejadores de bases de datos para la ingeniería de datos.

La ingeniería de datos es un campo interdisciplinario que se enfoca en la recopilación, almacenamiento, procesamiento y análisis de datos para obtener información valiosa. Los manejadores de bases de datos desempeñan un papel crucial en este proceso al proporcionar una estructura organizada para almacenar datos y permitir un acceso eficiente a ellos.

Los DBMS proporcionan una serie de funciones que permiten a los ingenieros de datos realizar tareas como:

- **Administrar la estructura de la base de datos**: Los DBMS permiten a los ingenieros de datos definir el esquema de la base de datos, que es la estructura que define cómo se almacenan los datos.
- **Insertar, actualizar y eliminar datos**: Los DBMS proporcionan funciones para insertar, actualizar y eliminar datos de la base de datos.
- **Realizar consultas a la base de datos**: Los DBMS proporcionan un lenguaje de consulta que permite a los ingenieros de datos extraer información de la base de datos.
- **Administrar el rendimiento de la base de datos**: Los DBMS proporcionan funciones para optimizar el rendimiento de la base de datos.

Tipos de Manejadores de Bases de Datos

- **Bases de Datos Relacionales**: Estas bases de datos utilizan tablas para organizar los datos y se basan en el modelo relacional. Ejemplos populares incluyen MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Son ampliamente utilizados en aplicaciones empresariales y de análisis de datos.
- **Bases de Datos No Relacionales (NoSQL):** Estos sistemas de bases de datos están diseñados para manejar datos no estructurados o semiestructurados y son flexibles en términos de esquema. Ejemplos incluyen MongoDB, Cassandra, Redis y Elasticsearch. Son ideales para casos de uso como aplicaciones web y Big Data.
- **Bases de Datos Columnares**: Estas bases de datos almacenan datos en columnas en lugar de filas, lo que las hace eficientes para operaciones de análisis y consultas de agregación. Ejemplos incluyen Apache Cassandra y Amazon Redshift.
- **Bases de Datos en Memoria**: Estas bases de datos almacenan datos en la memoria RAM en lugar de en discos, lo que permite un acceso extremadamente rápido a los datos. Ejemplos incluyen Redis y Memcached.

Características Clave

- **Escalabilidad**: Deben ser capaces de manejar grandes volúmenes de datos y escalarse horizontal o verticalmente según sea necesario.
- **Velocidad**: Deben ofrecer un rendimiento rápido en la inserción y recuperación de datos, especialmente en entornos de Big Data.
- **Durabilidad**: Los datos deben persistir de manera confiable incluso en caso de fallos del sistema.
- **Seguridad**: Deben proporcionar mecanismos de autenticación y autorización para proteger los datos sensibles.
- **Facilidad de Uso**: Deben tener interfaces de administración y consulta intuitivas.
- **Compatibilidad con Lenguajes de Programación**: Deben ser compatibles con una variedad de lenguajes de programación para facilitar el desarrollo de aplicaciones.

Características Clave Arquitectura

Los manejadores de bases de datos pueden ser instalados en un solo sitio o en múltiples sitios, según la distribución de los procesos y los datos. Esto implica diferentes niveles de complejidad y rendimiento,

- **Centralizados**: se ejecutan en una sola computadora o servidor, donde se almacenan todos los datos. Esta arquitectura es simple y fácil de administrar, pero también tiene limitaciones en cuanto a la capacidad, la disponibilidad y la seguridad de los datos.
- **Distribuidos**: que se ejecutan en varias computadoras o servidores, donde se almacenan partes o réplicas de los datos. Esta arquitectura es más compleja y difícil de administrar, pero también ofrece mayores beneficios en cuanto a la escalabilidad, la tolerancia a fallos y el acceso a los datos.
- **Cliente-servidor**: que se ejecutan en dos o más computadoras o servidores, donde una actúa como cliente y otra como servidor. El cliente se encarga de enviar las solicitudes al servidor, y el servidor se encarga de procesarlas y devolver los resultados al cliente. Esta arquitectura es muy común y permite separar las funciones del sistema, mejorar el rendimiento y facilitar la comunicación entre los usuarios.

Universidad de la Ciudad de Aguascalientes

Tendencias en Manejadores de Bases de Datos para Ingeniería de Datos

- **Bases de Datos en la Nube**: La migración hacia bases de datos en la nube, como Amazon RDS, Google Cloud SQL y Azure SQL DataBase, está en aumento debido a la escalabilidad y la facilidad de administración que ofrecen.
- **Bases de Datos Distribuidas**: El uso de bases de datos distribuidas, como Apache Cassandra y Hadoop HBase, es esencial para el procesamiento de datos distribuidos y la escalabilidad masiva.
- **Bases de Datos de Gráficos**: Las bases de datos de gráficos, como Neo4j, son cada vez más importantes para aplicaciones que requieren análisis de relaciones complejas.
- **Integración con Big Data**: Los manejadores de bases de datos se integran cada vez más con herramientas de Big Data, como Hadoop y Spark, para realizar análisis avanzados.

Nota: los manejadores de bases de datos desempeñan un papel esencial en la ingeniería de datos al proporcionar la infraestructura necesaria para almacenar y acceder a los datos de manera eficiente. La elección del tipo de base de datos adecuado depende de los requisitos específicos del proyecto y de las características de los datos que se están manejando. Con el continuo avance de la tecnología, se esperan más innovaciones en este campo en el futuro.

Manejadores de bases de datos **no relacionales** para ingeniería de datos

- **MongoDB**: Es un DBMS no relacional de código abierto que es popular por su escalabilidad y su flexibilidad.
- **Cassandra**: Es un DBMS no relacional de código abierto que es popular por su fiabilidad y su disponibilidad.
- **Neo4j**: Es un DBMS no relacional de código abierto que es popular para el análisis de redes.
- **HBase**: Es un DBMS no relacional de código abierto que es popular para el almacenamiento de datos de gran volumen.

Manejadores de bases de datos **relacionales** para ingeniería de datos

- **MySQL**: Es un DBMS de código abierto que es popular por su facilidad de uso y su escalabilidad.
- **PostgreSQL**: Es un DBMS de código abierto que es popular por su rendimiento y su compatibilidad con estándares.
- **Oracle DB**: Es un DBMS comercial que es popular por su fiabilidad y su seguridad.
- **Microsoft SQL Server**: Es un DBMS comercial que es popular por su integración con las aplicaciones de Microsoft.

Tips & Tricks: Elección del manejador de bases de datos

La elección del manejador de bases de datos adecuado para un proyecto de ingeniería de datos depende de una serie de factores, entre los que se incluyen:

- El **tipo de datos que se almacenarán** en la base de datos: Los manejadores de bases de datos relacionales son adecuados para almacenar datos estructurados, mientras que los manejadores de bases de datos no relacionales son adecuados para almacenar datos semiestructurados o no estructurados.

- El **tamaño** de la base de datos: Los manejadores de bases de datos relacionales pueden escalar a bases de datos de gran tamaño, mientras que los manejadores de bases de datos no relacionales pueden ser más adecuados para bases de datos de tamaño mediano o pequeño.

- Los requisitos de **rendimiento**: Los manejadores de bases de datos relacionales pueden ser más rápidos que los manejadores de bases de datos no relacionales para las consultas simples, mientras que los manejadores de bases de datos no relacionales pueden ser más rápidos para las consultas complejas.

- Los requisitos de **seguridad**: Los manejadores de bases de datos comerciales suelen ofrecer más funciones de seguridad que los manejadores de bases de datos de código abierto.

Referencias:

- Mukherjee, Sourav. (2019). The battle between NoSQL Databases and RDBMS. 10.15680/IJIRSET.2019.0805107.
- Bodepudi, Hariteja. (2020). Faster The Slow Running RDBMS Queries With Spark Framework. International Journal of Scientific and Research Publications (IJSRP). 10. 10.29322/IJSRP.10.11.2020.p10735.
- Sabiguero, Ariel & Etcheverry, Lorena & Vicente, Alfonso. (2021). An RDBMS-only architecture for web applications. 10.1109/CLEI53233.2021.9640017.
- Bansal, Neha & Soni, Kanika & Sachdeva, Shelly. (2022). Journey of Database Migration from RDBMS to NoSQL Data Stores. 10.1007/978-3-030-96600-3_12.
- Zeng, Li & Zhou, Jinhua & Qin, Shijun & Cai, Haoran & Zhao, Rongqian & Chen, Xin. (2022). SQLG+: Efficient k-hop Query Processing on RDBMS. 10.1007/978-3-031-00129-1_37.
- Tripathi, Harish Kumar & Jain, Vivekanand. (2023). ANALYTICAL STUDY OF BACK END RDBMS FOR SERVER DATABASE IN LIBRARY AUTOMATION ENVIRONMENT.
- Vicente, Alfonso. (2022). La arquitectura RDBMS-only. Una arquitectura database-centric para aplicaciones Web.
- Chaturvedi, Arpana & Khanna, Deepti. (2023). Optimization and Pursuance Analysis of RDBMS for Relational Algebraic Operations—MySQL. 10.1007/978-981-19-2065-3_34.
- Akter, Shapla & Hossain, Md & Hossain, Shahadat & Ghosh, Joyanta & Dehan, Md & Hasan, Md. (2022). Is RDBMS or NoSQL Better Suited for MIS?: A Comparative Analysis. 10.1007/978-3-031-19958-5_106.
- Hahn, Sarah & Chereja, Ionela & Matei, Oliviu. (2023). Analysis of the Performance of NewSQL Databases Compared to RDBMS Based on Linux OS. 10.1007/978-3-031-21435-6_59.
- Arshad, Muhammad & Brohi, M. & Soomro, Tariq & Ghazal, Taher & Alzoubi, Haitham & Alshurideh, Muhammad. (2023). NoSQL: Future of BigData Analytics Characteristics and Comparison with RDBMS. 10.1007/978-3-031-12382-5_106.
- Abbas, Sameera & Jameel, Enas. (2022). A COMPARISON BETWEEN NOSQL AND RDBMS: STORAGE AND RETRIEVAL. MINAR International Journal of Applied Sciences and Technology. 04. 172-184. 10.47832/2717-8234.12.18.

3. Desarrollo

3.1 Lenguajes para el desarrollo de software
3.2 Manejadores de bases de datos
3.3 Construcción del sistema

La construcción del sistema en ingeniería de datos es un proceso esencial en la gestión y análisis de datos. Implica la creación de una infraestructura y un entorno que permitan la recopilación, almacenamiento, procesamiento y análisis eficiente de datos para satisfacer las necesidades de una organización. A continuación, se presenta una investigación sobre este tema.

- La construcción del sistema en ingeniería de datos es un proceso multidisciplinario que abarca la recopilación, el almacenamiento, el procesamiento y la gestión de datos. Es fundamental para habilitar la toma de decisiones basadas en datos en las organizaciones y está en constante evolución para adaptarse a las cambiantes necesidades tecnológicas y empresariales.

- La correcta construcción del sistema de ingeniería de datos requiere una comprensión profunda de las necesidades de la organización, una selección adecuada de tecnologías y una atención constante a la seguridad y el rendimiento de los datos. Además, las tendencias emergentes, como el aprendizaje automático y la nube, están transformando la forma en que se aborda este proceso fundamental en la actualidad.

Nota: la construcción del sistema en ingeniería de datos es un campo dinámico y esencial que impulsa la capacidad de las organizaciones para utilizar sus datos de manera efectiva en un mundo cada vez más impulsado por la información.

Componentes clave de la construcción del sistema en ingeniería de datos:

- **Recopilación de datos**: El primer paso es recopilar datos de diversas fuentes, que pueden incluir bases de datos, sistemas de registro, sensores, aplicaciones web y más. Esto a menudo implica la integración de sistemas dispares para reunir datos en un solo lugar.
- **Almacenamiento de datos**: Una vez recopilados, los datos deben almacenarse adecuadamente. Las tecnologías de almacenamiento de datos incluyen bases de datos relacionales, almacenes de datos, sistemas de archivos distribuidos y soluciones de almacenamiento en la nube.
- **Procesamiento de datos**: Los datos a menudo requieren limpieza, transformación y agregación antes de ser útiles. Se utilizan herramientas como Apache Spark, Hadoop y ETL (Extract, Transform, Load) para realizar estas tareas.
- **Modelado de datos**: En esta etapa, se diseñan y desarrollan modelos de datos que representen de manera eficiente la información que se desea analizar. Esto incluye la definición de esquemas de bases de datos y la creación de cubos OLAP, si es necesario.
- **Seguridad de datos**: La seguridad de los datos es fundamental en la construcción del sistema en ingeniería de datos. Se deben establecer controles de acceso y cifrado para proteger la confidencialidad e integridad de los datos.
- **Gestión de metadatos**: Los metadatos son datos que describen otros datos. La gestión de metadatos ayuda a entender y documentar la estructura y el significado de los datos, facilitando su uso y mantenimiento.
- **Automatización y orquestación**: La automatización de procesos y la orquestación de flujos de datos permiten ejecutar tareas de forma programada y coordinada, lo que mejora la eficiencia y la consistencia.
- **Monitoreo y gestión de rendimiento**: Es esencial supervisar el rendimiento del sistema en tiempo real y realizar ajustes según sea necesario para garantizar un rendimiento óptimo y resolver problemas rápidamente.

Tendencias en la construcción del sistema en ingeniería de datos:

- **Aprendizaje automático y AI**: La integración de algoritmos de aprendizaje automático y de inteligencia artificial en el proceso de ingeniería de datos está en auge, lo que permite obtener información más profunda y automatizar tareas de análisis.
- **Soluciones en la nube**: Cada vez más, las organizaciones están migrando sus sistemas de ingeniería de datos a la nube para aprovechar la escalabilidad y la flexibilidad que ofrece.
- **Procesamiento en tiempo real**: La capacidad de procesar datos en tiempo real se ha vuelto esencial en aplicaciones como el análisis de redes sociales y la detección de fraudes.
- **Gestión de datos de extremo a extremo**: Las soluciones de gestión de datos de extremo a extremo, que abarcan desde la recopilación hasta el análisis y la visualización, están ganando terreno.

Referencias.

- Almeida, Silvia & Dávila, Abraham. (2023). A Systematic Mapping Study on Process Improvement in Software Requirements Engineering. Proceedings of the Institute for System Programming of the RAS. 35. 141-162. 10.15514/ISPRAS-2023-35(1)-10.

- Tosun, Ayse & Bener, Ayse & Caglayan, Bora & Calikli, Gul & Turhan, Burak. (2014). Field Studies in the Construction and Evaluation of Recommendation Systems in Software Engineering.

- B., Sreejith & V., Sreeja. (2023). The Construction Agile Managements of Different Infrastructure Projects. 10.59544/WKYJ3792/NGCESI23P105.

- Savary-Leblanc, Maxime & Burgueño, Lola & Cabot, Jordi & Pallec, Xavier & Gérard, Sébastien. (2022). Software assistants in software engineering: A systematic mapping study. Software: Practice and Experience. 53. 10.1002/spe.3170.