

**Universidad Tecnológica de Panamá**

**Facultad de Ingeniería en Sistemas Computacionales**

**Ingeniería en Sistemas de Información**

**Ingeniería de Software Aplicada**

**Investigación**

**Arquitectura de Software Centrada en Datos**

**Facilitadora**: Ing. Jeanette Riley

**Estudiante:**

**Johel Batista 8-914-587**

**Enlace de la Charla:** [**https://youtu.be/Cj0VDNw5sog**](https://youtu.be/Cj0VDNw5sog)

**I Semestre Académico**

Índice de Contenidos

[Introducción o Comentarios Iniciales 2](#_Toc137339740)

[¿Qué es la Arquitectura de Software? 4](#_Toc137339741)

[Arquitectura de Software Centrada en Datos 5](#_Toc137339742)

[Ventajas y Desventajas de la Arquitectura Centrada en Datos 7](#_Toc137339743)

[Ventajas 7](#_Toc137339744)

[Desventajas 8](#_Toc137339745)

[Repositorios 8](#_Toc137339746)

[Repositorios de Datos 9](#_Toc137339747)

[Repositorios de Metadatos 9](#_Toc137339748)

[Repositorios de Código 10](#_Toc137339749)

[Repositorios Distribuidos 10](#_Toc137339750)

[Conclusión o Comentarios Finales 12](#_Toc137339751)

[Referencias Bibliográficas 13](#_Toc137339752)

# Introducción o Comentarios Iniciales

En el vasto universo de la ingeniería de software, la arquitectura de software se presenta como una disciplina crítica que proporciona una estructura sólida y cohesiva para el desarrollo de sistemas complejos. Es la concepción de alto nivel de un sistema de software, el esqueleto que determina cómo se organizan y cómo interactúan los componentes. A través de la arquitectura de software, los ingenieros pueden dirigir el flujo de datos, la distribución de tareas y la interacción de subsistemas, garantizando al mismo tiempo la eficiencia, la escalabilidad y la seguridad del sistema.

En el contexto de la diversidad de arquitecturas de software, las Arquitecturas de Software Centradas en Datos se destacan por su enfoque en la gestión y manipulación de los datos. Estas arquitecturas reconocen que los datos son el recurso más valioso y la fuente principal de la funcionalidad del sistema. En estas arquitecturas, los datos no son solo un componente pasivo que se procesa, sino que son la entidad central alrededor de la cual se construye todo el sistema.

Las Arquitecturas de Software Centradas en Datos se basan en la idea de que los datos deben ser accesibles y manejables, permitiendo a los sistemas manejar grandes volúmenes de información de manera eficiente y confiable. Se enfocan en cómo se almacenan, se accede, se manipulan y se mantienen los datos. Estos sistemas a menudo emplean técnicas y tecnologías como las bases de datos, los data warehouses, los sistemas de gestión de bases de datos y las tecnologías de big data para gestionar los datos.

En el mundo moderno, donde los datos son omnipresentes y cada vez más cruciales para las decisiones empresariales, la investigación en Arquitecturas de Software Centradas en Datos es más relevante que nunca. La necesidad de manejar grandes volúmenes de datos, de extraer valor de ellos y de garantizar su seguridad y privacidad pone a estas arquitecturas en el centro de la innovación y el avance tecnológico. Por tanto, nuestra investigación se centrará en profundizar en las diversas facetas de las Arquitecturas de Software Centradas en Datos, explorando sus principios, su implementación, sus desafíos y sus aplicaciones en el mundo real.

A medida que avanzamos, nos sumergiremos en el rico universo de las Arquitecturas de Software Centradas en Datos, abriendo camino hacia una comprensión más profunda de su rol, su valor y su potencial en la ingeniería de software. Nos embarcaremos en una travesía de descubrimiento, buscando arrojar luz sobre la manera en que estas arquitecturas moldean y serán moldeadas por el paisaje tecnológico en constante evolución.

# ¿Qué es la Arquitectura de Software?

La arquitectura de software es un tema complejo y multifacético que implica una serie de conceptos interrelacionados. En su esencia, la arquitectura de software puede ser vista como el plan maestro de un sistema.

Es el marco dentro del cual se diseñan, se desarrollan y se integran los componentes de un sistema de software. Una de las primeras partes de la arquitectura de software es el diseño arquitectónico. Este es el proceso de definir y describir los componentes de alto nivel del sistema y sus interrelaciones.

Involucra la toma de decisiones estratégicas sobre la organización general del sistema, incluyendo la descomposición del sistema en subsistemas y módulos, y la definición de las interfaces y la interacción entre ellos.

Los componentes son entidades de software independientes que realizan una función específica dentro del sistema. Pueden ser simples o complejos, e incluyen cosas como módulos, clases, objetos, y subsistemas.

Los componentes son la unidad básica de la modularidad en la arquitectura de software, y su correcta identificación, diseño e implementación son cruciales para el éxito del sistema. Las interfaces son otro elemento clave de la arquitectura de software. Una interfaz define cómo un componente se comunica con otros componentes y cómo se espera que interactúe con ellos.

Las interfaces proporcionan abstracción, permitiendo que los componentes interactúen sin necesidad de conocer los detalles internos de cada uno. Los patrones de arquitectura, también conocidos como estilos arquitectónicos, son soluciones probadas a problemas comunes de diseño de software. Ejemplos de estilos arquitectónicos son la arquitectura cliente-servidor, la arquitectura en capas, la arquitectura basada en eventos, y la arquitectura de microservicios.

La elección del estilo arquitectónico adecuado es fundamental, ya que tiene un impacto significativo en las propiedades y cualidades del sistema. Las cualidades del sistema, también conocidas como atributos de calidad o características no funcionales, son propiedades globales del sistema que afectan a su comportamiento y rendimiento.

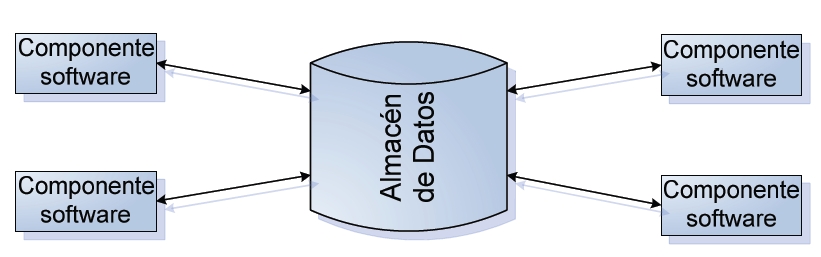
Incluyen cosas como la seguridad, la escalabilidad, la fiabilidad, la usabilidad, y la mantenibilidad. La arquitectura de software juega un papel crucial en la obtención de estas cualidades. Por último, pero no menos importante, la documentación de la arquitectura es un aspecto esencial de la arquitectura de software.

# Arquitectura de Software Centrada en Datos

**Es un** enfoque en el diseño de sistemas de software donde los datos y su gestión se consideran el aspecto central del sistema.

Este paradigma reconoce la importancia creciente de los datos en el mundo digital y busca proporcionar estructuras y métodos eficientes para su manejo, procesamiento y análisis. En primer lugar, es importante entender qué se entiende por "datos". Los datos se refieren a hechos, estadísticas, o información recopilada para su análisis y uso en la toma de decisiones.

En el contexto de una Arquitectura de Software Centrada en Datos, estos pueden ser datos estructurados (como los que se encuentran en bases de datos relacionales), datos semi-estructurados (como los datos en XML o JSON), o datos no estructurados (como los datos de texto, audio o video).



***Ilustración #1****: Diagrama General de la Arquitectura Centrada en Datos*

El primer concepto clave en la Arquitectura de Software Centrada en Datos es el de la persistencia de datos. Esto se refiere a la capacidad de almacenar datos de manera que perduren más allá de la duración de un programa o sesión específica.

En la mayoría de los casos, esto implica el uso de algún tipo de base de datos, que puede ser relacional (como MySQL o PostgreSQL), NoSQL (como MongoDB o Cassandra), o una base de datos de series temporales (como InfluxDB), dependiendo de las necesidades específicas de los datos y la aplicación.

Además, la gestión de los datos es un aspecto fundamental de estas arquitecturas. Esto puede incluir la ingesta de datos (la recolección y aceptación de datos en el sistema), la limpieza y transformación de datos (asegurándose de que los datos estén en un formato utilizable y sean de alta calidad), y la indexación de datos (haciendo que los datos sean rápidamente buscables).

También es importante destacar el papel de las tecnologías de Big Data en la Arquitectura de Software Centrada en Datos. Herramientas como Hadoop y Spark permiten el procesamiento de grandes volúmenes de datos distribuidos a través de múltiples nodos, lo cual es esencial para muchas aplicaciones modernas que manejan enormes cantidades de datos.

Otros sistemas como Kafka o RabbitMQ pueden ser utilizados para la transmisión de datos en tiempo real entre componentes del sistema.Finalmente, una Arquitectura de Software Centrada en Datos también necesita considerar la seguridad y privacidad de los datos. Esto puede implicar la encriptación de datos, la gestión de derechos de acceso, y la implementación de políticas de privacidad y cumplimiento.

Sin embargo, su enfoque en los datos como el núcleo del sistema de software la convierte en una opción poderosa para muchas aplicaciones modernas, especialmente en un mundo donde los datos son cada vez más valiosos y omnipresentes**.**

# Ventajas y Desventajas de la Arquitectura Centrada en Datos

La Arquitectura de Software Centrada en Datos es un enfoque que tiene tanto ventajas como desventajas, dependiendo del contexto y las necesidades específicas del sistema.

## Ventajas

1. **Mejora la consistencia y la calidad de los datos:** Al centrarse en los datos como el elemento principal, se enfatiza la necesidad de mantener la integridad y la consistencia de los datos en todo el sistema. Esto puede llevar a una mejor calidad de los datos, ya que se realizan esfuerzos para asegurar que los datos sean precisos, completos y actualizados.
2. **Aumento de la eficiencia en el manejo de los datos**: Las tecnologías y técnicas utilizadas en este tipo de arquitectura, como las bases de datos y los sistemas de gestión de bases de datos, están optimizadas para el manejo eficiente de los datos. Esto puede llevar a un mejor rendimiento y a una mayor eficiencia en el uso de recursos.
3. **Mejora la reutilización de los datos**: Al mantener los datos en un lugar centralizado, se facilita su reutilización en diferentes partes del sistema o en diferentes aplicaciones. Esto puede llevar a una mayor eficiencia y a la reducción de la redundancia de los datos.
4. **Facilita la integración de los datos:** En un mundo donde los datos provienen de una variedad de fuentes y formatos, una Arquitectura de Software Centrada en Datos puede facilitar la integración de estos datos heterogéneos, proporcionando un marco unificado para su gestión.

## Desventajas

1. **Puede llevar a la dependencia de los datos**: Al centrarse tanto en los datos, puede surgir una dependencia excesiva de ellos. Si los datos se corrompen o se pierden, puede tener un impacto significativo en todo el sistema.
2. **Problemas de escalabilidad**: A medida que la cantidad de datos crece, puede ser un desafío mantener el rendimiento y la eficiencia del sistema. Aunque las tecnologías modernas de Big Data pueden ayudar a abordar este problema, la escalabilidad sigue siendo un desafío en las arquitecturas centradas en datos.
3. **Problemas de seguridad y privacidad**: Los datos son a menudo sensibles y valiosos, y su centralización puede hacerlos un objetivo atractivo para los ataques. Además, la gestión de los derechos de acceso a los datos puede ser compleja en un sistema centrado en datos.
4. **Necesidad de habilidades y conocimientos especializados**: La gestión eficaz de los datos requiere una comprensión profunda de una variedad de técnicas y tecnologías, incluyendo bases de datos, lenguajes de consulta de datos, y tecnologías de Big Data. Esto puede aumentar la complejidad del sistema y requerir habilidades y conocimientos especializados.

La elección de este enfoque dependerá de las necesidades específicas del sistema y del contexto en el que se utilice.

# Repositorios

En la Arquitectura de Software Centrada en Datos, los repositorios desempeñan un papel crucial, ya que son los componentes responsables de la gestión y el almacenamiento de los datos en el sistema.

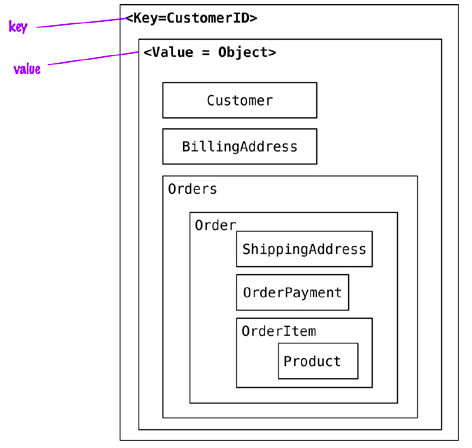
Un repositorio puede ser entendido como un contenedor de almacenamiento de datos que proporciona una serie de operaciones para acceder, modificar, y manipular los datos.

## ****Repositorios de Datos****

Los repositorios de datos son una forma especializada de repositorio que está optimizada para el almacenamiento y la gestión de grandes volúmenes de datos.

Estos suelen ser implementados utilizando alguna forma de sistema de gestión de bases de datos, ya sea relacional, NoSQL, o de otro tipo.

Los repositorios de datos proporcionan operaciones para la inserción, la consulta, la actualización y la eliminación de datos, y a menudo incluyen soporte para transacciones y la integridad de los datos.



***Imagen #2****: Ejemplo de una Base de Datos tipo Clave-Valor*

## ****Repositorios de Metadatos****

Los repositorios de metadatos se utilizan para almacenar metadatos, que son datos acerca de los datos.

Los metadatos pueden incluir información como el esquema de los datos, las relaciones entre los datos, la proveniencia de los datos, y las políticas de acceso a los datos.

Los repositorios de metadatos son esenciales para la gestión de los datos en una Arquitectura de Software Centrada en Datos, ya que proporcionan el contexto necesario para entender y utilizar correctamente los datos.

## ****Repositorios de Código****

En el contexto de la Arquitectura de Software Centrada en Datos, los repositorios de código también pueden ser importantes.

Estos son repositorios que almacenan el código fuente del sistema, incluyendo los algoritmos y las funciones utilizadas para procesar y analizar los datos. Los repositorios de código proporcionan un medio para la gestión de versiones, la colaboración, y la trazabilidad del código.



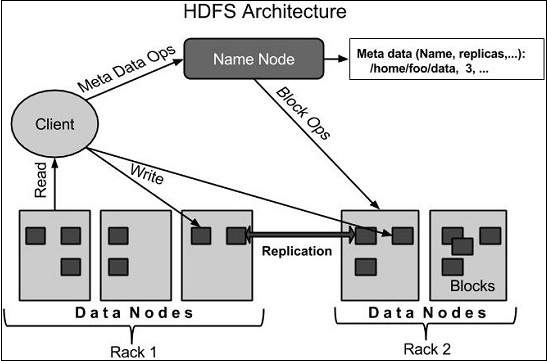
***Imagen #3****: Ejemplos de Repositorio de Código más utilizados*

## ****Repositorios Distribuidos****

Con la creciente necesidad de manejar grandes volúmenes de datos, los repositorios distribuidos se han vuelto cada vez más comunes.

Estos son repositorios que dividen los datos en múltiples nodos en una red, permitiendo el almacenamiento y procesamiento paralelo de los datos.

Los sistemas de archivos distribuidos como Hadoop Distributed File System (HDFS) y las bases de datos distribuidas como Cassandra son ejemplos de repositorios distribuidos.



***Imagen #4****: Arquitectura Básica de una implementación de HDFS*

Cada tipo de repositorio tiene sus propias ventajas, desventajas, y casos de uso, y la elección del tipo de repositorio apropiado dependerá de las necesidades específicas del sistema.

# Conclusión o Comentarios Finales

La arquitectura centrada en datos ha emergido como un enfoque crucial en el desarrollo de software, debido a la creciente relevancia de los datos en la toma de decisiones y la generación de valor en las organizaciones.

Se ha establecido que esta arquitectura se centra en el manejo y procesamiento de los datos, facilitando su acceso y manejo a través de diversos componentes del sistema.

Un elemento fundamental en la arquitectura centrada en datos es el repositorio de datos, que es esencialmente una ubicación central para almacenar y manejar datos. Hemos analizado varios tipos de repositorios, incluyendo bases de datos operativas, data warehouses, y data marts, cada uno con sus propias características y aplicaciones.

También se ha hecho hincapié en la importancia de las técnicas y tecnologías de manejo de datos, como los sistemas de gestión de bases de datos (DBMS), que son fundamentales para el almacenamiento, recuperación y manipulación de datos.

En términos de las ventajas de la arquitectura centrada en datos, hemos resaltado cómo puede proporcionar coherencia y eficiencia en el manejo de datos, al permitir un acceso uniforme y centralizado a los datos.

También puede facilitar la integración de sistemas y la interoperabilidad, al permitir que diferentes componentes del sistema accedan y trabajen con los mismos datos. Sin embargo, también hemos señalado algunas de las posibles desventajas, como la necesidad de una gestión y gobernanza de datos eficaz para garantizar la calidad y seguridad de los datos.

A lo largo del estudio, también hemos discutido varios conceptos y tecnologías clave que van de la mano con la arquitectura centrada en datos, incluyendo la integración de datos, el procesamiento de transacciones en línea (OLTP), el procesamiento analítico en línea (OLAP), el Big Data y las tecnologías de la nube.

Estos conceptos y tecnologías son fundamentales para entender y trabajar con arquitecturas centradas en datos.

En resumen, la arquitectura de software centrada en datos es un enfoque potente y versátil para el desarrollo de software que puede proporcionar numerosos beneficios en términos de eficiencia, coherencia y capacidad de integración.

Sin embargo, también requiere una gestión y gobernanza de datos eficaz para garantizar la calidad y seguridad de los datos.

A medida que el papel de los datos continúa creciendo en importancia, es probable que veamos un uso cada vez mayor de la arquitectura centrada en datos en diversas áreas del desarrollo de software.

# Referencias Bibliográficas

1. López Martínez, P., Dintén, R., Drake, J. M., & Zorrilla, M. (2021). A big data-centric architecture metamodel for Industry 4.0. Future Generation Computer Systems, 125, 263-284. Recuperado de [https://doi.org/10.1016/j.future.2021.06.020&#8203;``oaicite:{"number":1,"metadata":{"title":"A](https://doi.org/10.1016/j.future.2021.06.020&#8203;%60%60oaicite:%7B%22number%22:1,%22metadata%22:%7B%22title%22:%22A)
2. Avci, C., Tekinerdogan, B., & Athanasiadis, I. N. (2020). Software architectures for big data: a systematic literature review. Big Data Analytics, 5, Article 5. Recuperado de [https://bdataanalytics.biomedcentral.com/articles/10.1186/s41044-020-00045-1&#8203;``oaicite:{"number":2,"metadata":{"title":"Software](https://bdataanalytics.biomedcentral.com/articles/10.1186/s41044-020-00045-1&#8203;%60%60oaicite:%7B%22number%22:2,%22metadata%22:%7B%22title%22:%22Software)
3. Hajinazar, N. (2021). Data-Centric and Data-Aware Frameworks for Fundamentally Efficient Data Handling in Modern Computing Systems. Recuperado de [https://ar5iv.org/abs/2109.05881&#8203;``oaicite:{"number":3,"metadata":{"title":"[2109.05881]](https://ar5iv.org/abs/2109.05881&#8203;%60%60oaicite:%7B%22number%22:3,%22metadata%22:%7B%22title%22:%22%5B2109.05881%5D)