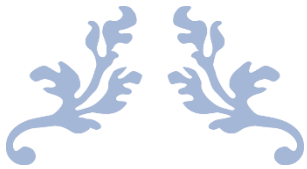


SISTEMAS DE BIG DATA

INVESTIGACIÓN PATRONES DE DISEÑO

Descripción Técnica

Los patrones de diseño son unas técnicas para resolver problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características.

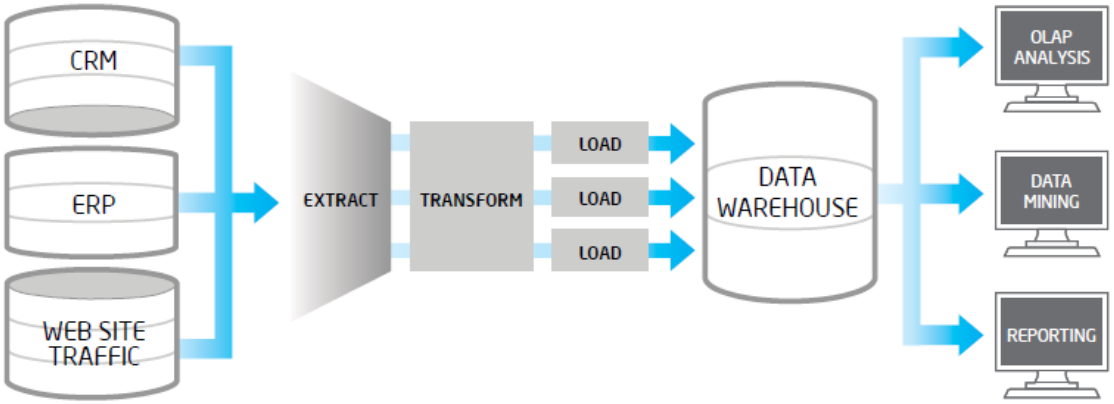
- Comprobar su efectividad a la hora de resolver problemas.
- Debe ser reutilizable.

Unos patrones de diseño que veremos a continuación serán:

- **ETL** - Extract, Transform, Load (Extraer, Transformar, Cargar).
- **ELT** - Extract, Load, Transform (Extraer, Cargar, Transformar).
- **CQRS** - Command Query Responsibility Segregation (Separación de la responsabilidad de consultas y comandos).

ETL - ¿Cómo funciona?

Toma datos sin procesar, los transforma en un formato predeterminado y, a continuación, los carga en el almacenamiento de datos de destino.



ELT - ¿Cómo funciona?

Toma datos sin procesar, los carga en el almacenamiento de datos de destino y, a continuación, los transforma justo antes del análisis.

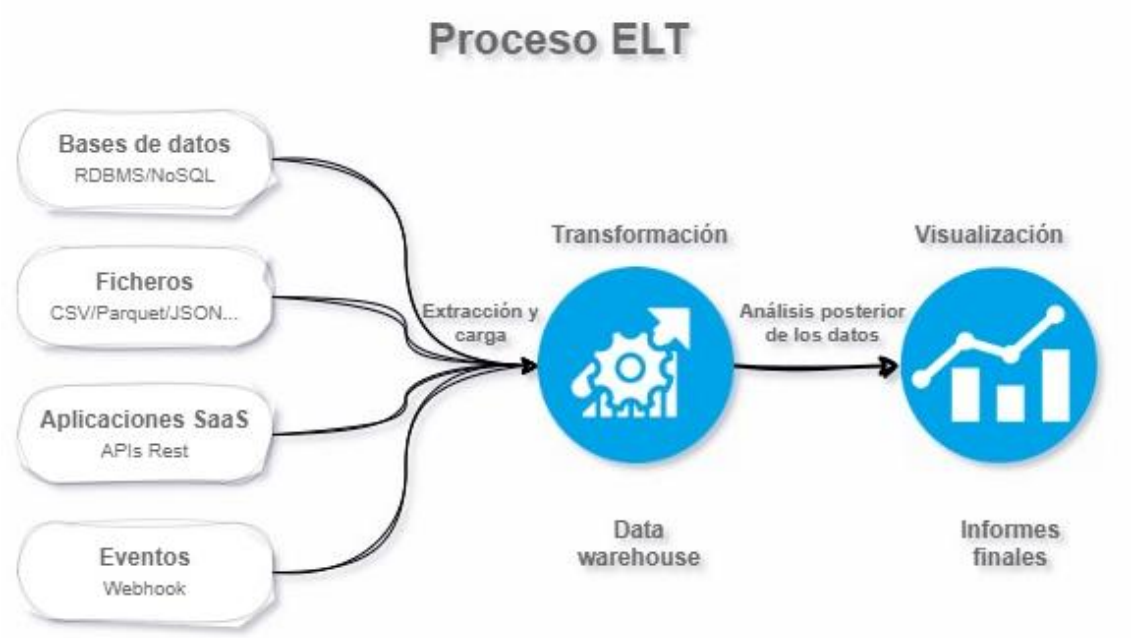
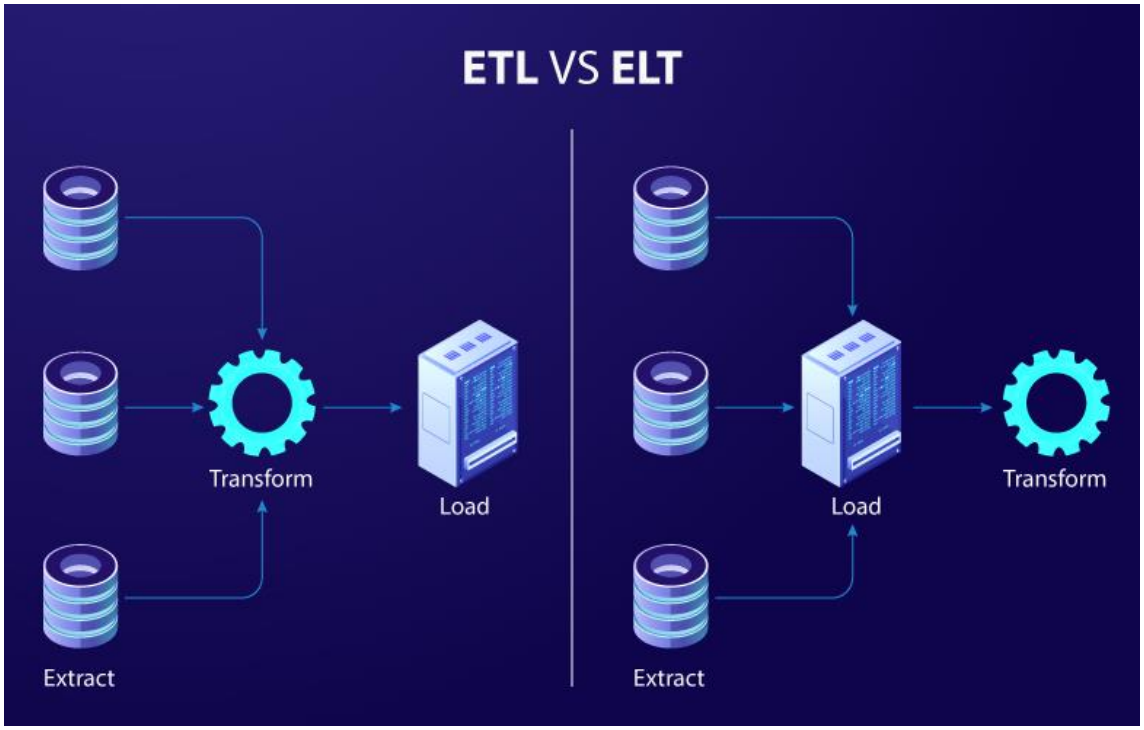


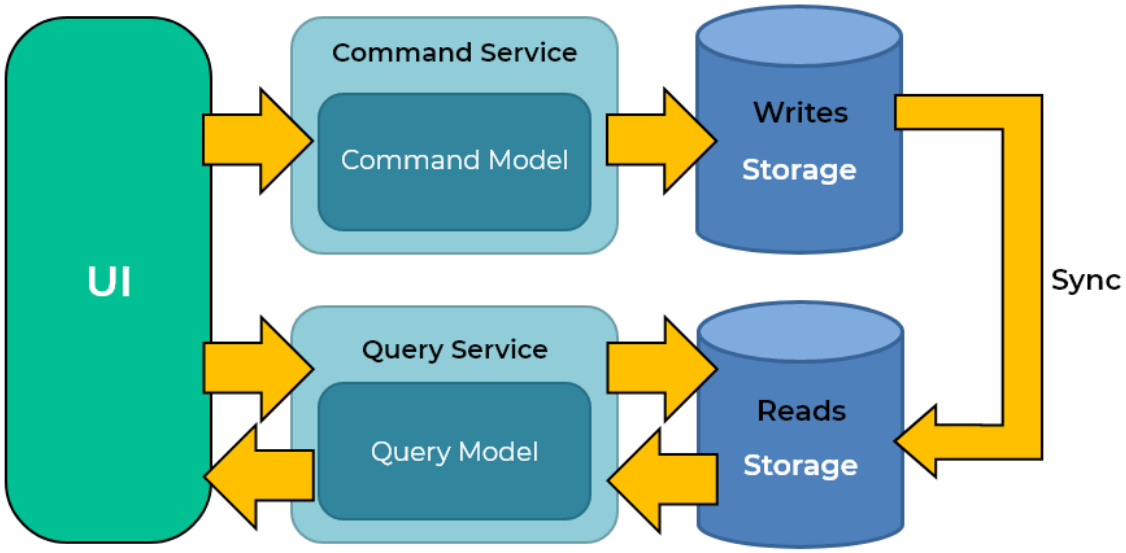
Imagen más simplificada del proceso ETL vs ELT:



CQRS – ¿Cómo funciona?

Tenemos una petición POST para crear un usuario en una API:

- El controlador recibirá los parámetros para la creación del usuario.
- Se crea un **Command** el cual será un **DTO** (objeto de transferencia de datos) con los parámetros de creación del usuario.
- El **CommandBus** recibe por parámetro el **Command** el cual se encarga de enviarlo al **CommandHandler.**
- El **CommnandHandler** recibirá el **Command** y este se encargará de enviarlo al caso de uso.
- El **caso de uso** validará los datos y los persistirá en la base de datos.



Command: Intención de realizar una acción en nuestro sistema que acabe modificando el estado como puede ser crear un registro, modificar uno existe o eliminarlo. El formato del Command será un DTO (objeto de transferencia de datos) el cual representa la acción que queremos hacer.

Query: representa la intención de solicitar datos a nuestro sistema sin que ello acabe alterando el estado de tal. Al igual que en el Command, una Query será un DTO el cual representará la petición de datos que queremos consultar.

Command/Query Bus: Este bus será el encargado de trasladar el DTO a su Handler correspondiente.

Command/Query Handler: En Handler recibirá el DTO y este lo enviará al caso de uso

Caso de Uso: Aquí es donde recibimos el DTO y aplicamos la lógica de negocio, validaremos los datos y después, según el tipo, los persistiremos o los recuperaremos.

Casos de Uso

El **patrón ETL** es más conocido y el **patrón ELT** es más moderno. Ambos son enfoques del procesamiento de datos que se utilizan para introducir datos en un almacén de datos y hacerlos útiles para los analistas y las herramientas de generación de reportes.

Algunas de las diferencias mas notables entre el patrón ETL y patrón ELT (quitando el orden de sus dos pasos finales) son:

- El proceso de ETL es más lento que el de ELT
- La configuración ETL puede llevar más tiempo y ser más costosa que la ELT (aunque depende de la infraestructura que se utilice).
- El patrón ETL se recomienda el uso de datos estructurados, el patrón ELT a datos estructurados, no estructurados y semiestructurados.

Ambos son ampliamente utilizados en almacenamiento de datos e inteligencia de negocios.

Respecto al **patrón CQRS** es un patrón de diseño de software que separa consultas (recuperar datos) de los comandos (inserción, actualización y borrado de datos). CQRS se utiliza en aplicaciones de alto rendimiento.

Ventajas y Desventajas

Patrón ETL	
Ventajas	**Desventajas**
Mejora la calidad de los datos	Tiempo alto de desarrollo.
Ahorra el tiempo necesario para mover, categorizar o estandarizar datos.	Rigidez, ya que se deben mapear todos los datos.
Optimización para las consultas.	Alterar cualquier paso en un flujo de trabajo de ETL puede romper otros flujos de trabajo.
Automatización y escalabilidad.	Altos costes.

Patrón ELT	
Ventajas	**Desventajas**
Flexibilidad para Transformaciones Ad-Hoc (solución específicamente elaborada para un problema).	La transformación puede ser costosa.
Mayor velocidad de carga de datos.	Riesgos de seguridad y dificulta el cumplimiento de las normas de privacidad de datos.
Aprovechamiento de recursos, lo que reduce la carga en servidores externos.	Posible sobrecarga de procesamiento si no se optimiza.

Patrón CQRS	
Ventajas	**Desventajas**
Permite seguir más de cerca el principio de responsabilidad única	Aumenta la complejidad del sistema
Si los subsistemas de escritura y lectura se separan físicamente, podrían escalarse de manera independiente	Problemas de transaccionalidad pueden llevar a incoherencias de datos, ya que un usuario podría consultar datos obsoletos
Las tecnologías de ambos sistemas podrían ser distintas: el sistema de escritura podría tener una base de datos distinta de la de lectura	Mayor esfuerzo en términos de desarrollo
Las consultas se vuelven más sencillas	Complicaciones en el sistema de mensajería

Tecnologías Asociadas

Patrón ETL

- **AWS Glue**: Es un servicio ETL completamente administrado que facilita la preparación y carga de datos en la nube.
- **AWS Data Pipeline**: Es un servicio de ETL administrado que permite definir el movimiento y las transformaciones de datos en varios servicios de AWS.
- **Microsoft SQL Server Integration Services (SSIS)**: Una solución de ETL de Microsoft usada para integraciones en entornos Windows.
- **Talend**: Es una solución de integración de datos de código abierto que ofrece una amplia gama de herramientas para ETL y ELT.
- **Google Cloud Dataflow:** Es una herramienta ETL completamente administrada, sin servidor y basada en eventos.
- **Apache Kafka:** Es una plataforma de streaming de eventos distribuidos que puede funcionar como una herramienta ETL en tiempo real.
- **Apache Nifi**: Es una herramienta ETL de código abierto diseñada para la automatización de flujos de datos entre sistemas.

Patrón ELT

- **Talend**: Es una solución de integración de datos de código abierto que ofrece una amplia gama de herramientas para ETL y ELT.
- **Snowflake**: Es una compañía de almacenamiento de datos basada en computación en la nube.
- **Google BigQuery**: Almacén de datos de Google que permite extraer analíticas de PB de datos.
- **Microsoft Azure Synapse**: Para manejar y cargar datos rápidamente, de la empresa Microsoft.
- **Apache Hadoop**: Es adoptada generalmente para gestionar grandes volúmenes y una gran variedad de datos.
- **Amazon Redshift Data API**: Simplifica el acceso al almacenamiento de datos.

Patrón CQRS

- **Netflix**: Para manejar la gran carga de tráfico y la cantidad de usuarios activos.
- **Microsoft Azure Cosmos DB**: Para manejar eventos y cargas masivas de datos de una manera eficiente y escalable.
- **Uber**: Para manejar los datos de transacciones y de los usuarios en su sistema.
- **Amazon**: Soportan el tráfico masivo de búsquedas y vistas de productos.
- **Ebay**: Para separar la lógica de negocio de consulta de las operaciones de compra y venta.

Bibliografía

Patrón de diseño

- https://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o
- https://refactoring.guru/es/design-patterns
- https://profile.es/blog/patrones-de-diseno-de-software/

ETL & ELT

- https://es.linkedin.com/advice/0/what-some-etl-design-patterns-automating-scheduling-5mdpe?lang=es
- https://forum.huawei.com/enterprise/es/Patrones-en-el-flujo-de-datos-ETL-ELT/thread/667238284230803456-667212895836057600
- https://nexla.com/data-integration-101/data-integration-architecture/
- https://aws.amazon.com/es/compare/the-difference-between-etl-and-elt/
- https://aws.amazon.com/es/what-is/etl/
- https://www.conectasoftware.com/magazine/las-16-mejores-herramientas-etl-para-2023/
- https://www.modus.es/etl-vs-elt/

- https://www.astera.com/es/type/blog/aws-etl-tools/#:~:text=Amazon%20Web%20Services%20(AWS)%20ETL,de%20decisiones%20basadas%20en%20ellos..
- https://learn.microsoft.com/es-es/azure/synapse-analytics/sql-data-warehouse/design-elt-data-loading
- https://www.stambia.com/es/productos/integracion-datos/para-proyectos-de-big-data-hadoop

CQRS

- https://cosasdedevs.com/posts/que-es-cqrs/
- https://docs.aws.amazon.com/prescriptive-guidance/latest/modernization-data-persistence/cqrs-pattern.html#:~:text=The%20command%20query%20responsibility%20segregation,throughput%2C%20latency%2C%20or%20consistency.
- https://learn.microsoft.com/en-us/azure/architecture/patterns/cqrs
- https://www.netmentor.es/entrada/patron-cqrs-explicado-10-minutos#mcetoc_1fui41i1boq
- https://www.altia.es/es/sobre-altia/actualidad/command-query-responsibility-segregation
- https://www.pragma.co/es/blog/patron-cqrs-que-es-y-como-implementarlo-dentro-del-framework-axon