Procesamiento de Datos Masivos

Clase 01 - Introducción

En este curso

En este curso, aprenderemos técnicas utilizadas para manejar grandes cantidades de datos, desde el punto de vista teórico y práctico

En esta clase

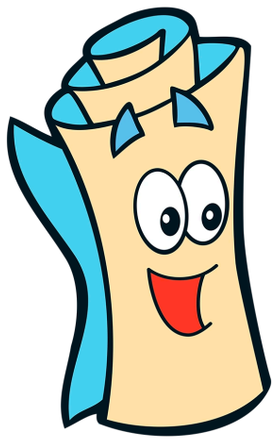
- Aprender sobre el ciclo de vida de los datos
- Entender el rol de un Data Engineer vs Data Scientist
- Entregar la noción sobre cómo armar un pipeline de datos

Recapitulemos un poco el viaje de trabajar con datos

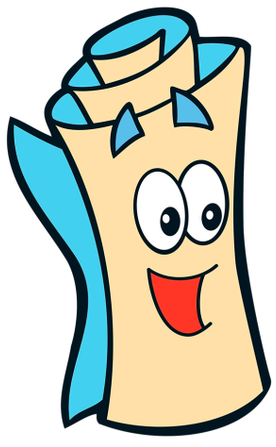

Quiero sacarle valor
a los datos

Pero no sé por donde partir :(

Pero no sé por donde partir :(



Soy el mapa, yo te ayudo!



En general, el ciclo de vida de los datos pasa por las siguientes etapas:

- El lago de datos
- El *data warehouse*
- *El procesamiento de datos*
- El análisis exploratorio de datos
- Modelos de Machine Learning
- Cómo evaluar nuestros modelos
- La puesta en producción

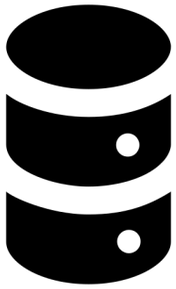
Un pipeline de datos mueve los datos a través de estas etapas

¿Por qué miramos los datos?

Una aplicación típica

En general, al crear una aplicación buscamos resolver un problema del mundo real

- Una aplicación para ahorrar dinero/invertir
- Una aplicación para hacer compras con envío a domicilio
- Una plataforma de aprendizaje en línea
- ...



Generamos datos de las transacciones de los usuarios

Análisis de datos

Cuando generamos datos, hay **conocimiento explícito** que podemos obtener a partir de los datos

En general, podemos obtener este conocimiento con un lenguaje de consultas como SQL

Ej. "Dame todos los usuarios que han comprado el producto **X** en el último mes"

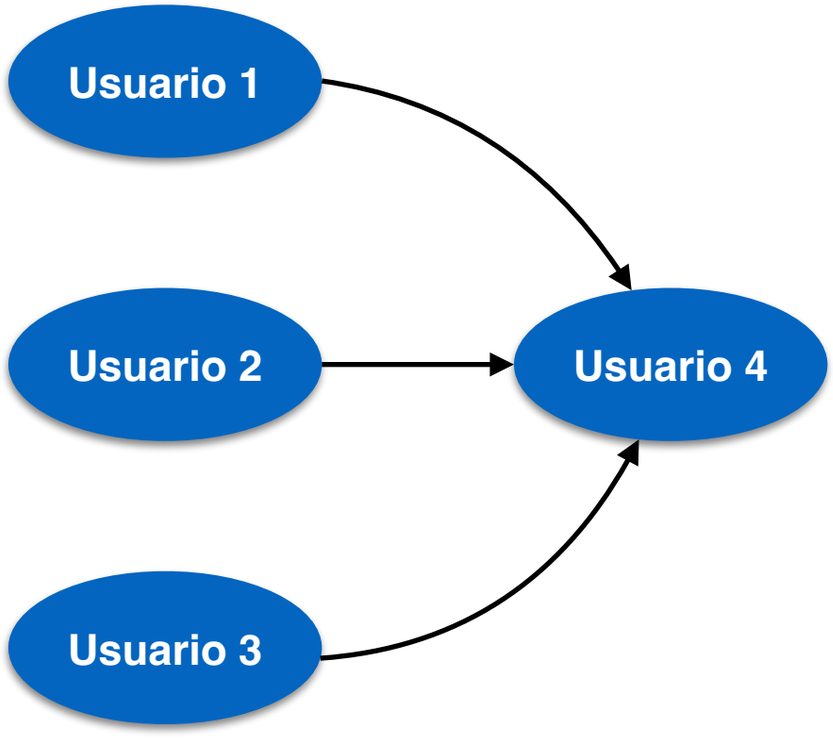
Análisis de datos

Pero hay mucho conocimiento que no está almacenado de forma explícita

Ese contenido lo vamos a extraer haciendo análisis de datos

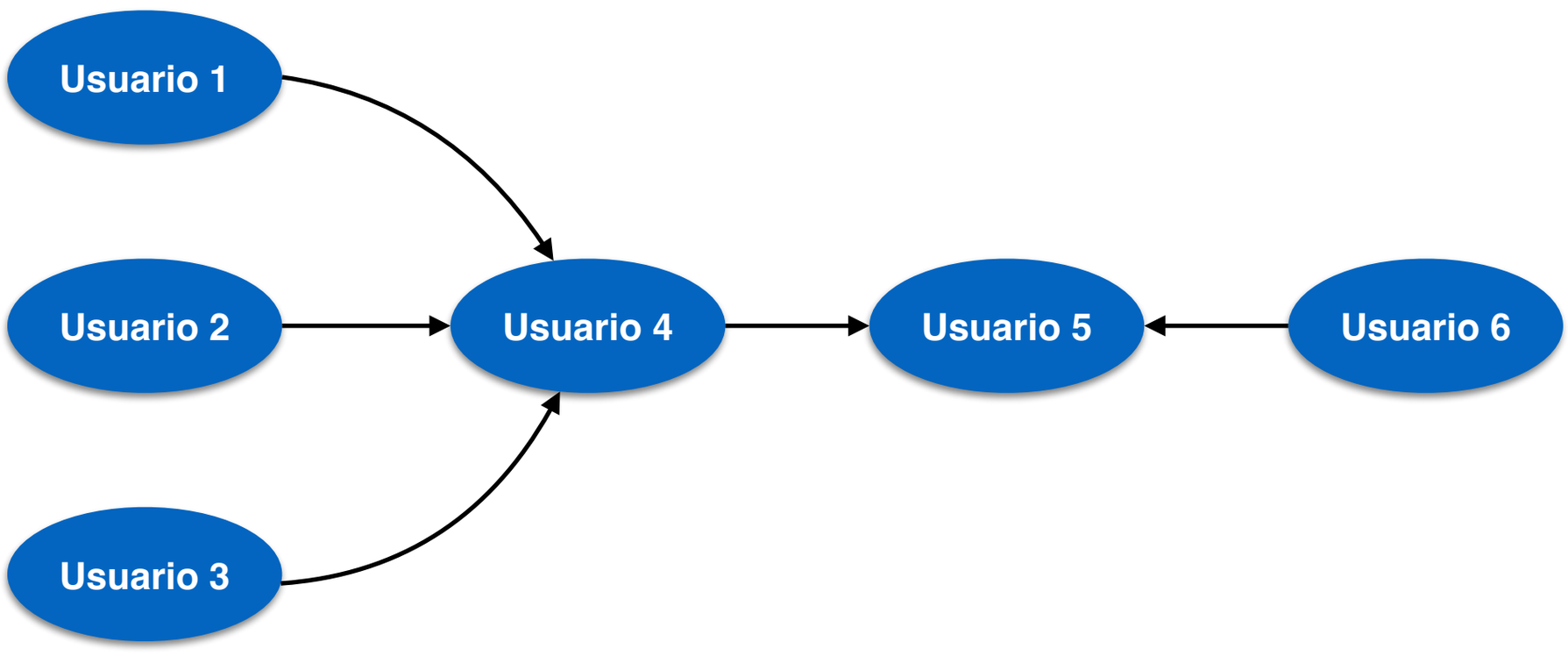
Ejemplo: medidas de centralidad

¿Cuál es el usuario más importante de esta red?



Ejemplo: medidas de centralidad

¿Y en esta red?



Análisis de datos

En el ejemplo anterior, podemos obtener mucha información de los datos explícitos

Pero saber el usuario más importante no es algo que podamos preguntar con un lenguaje de consultas

Análisis de datos

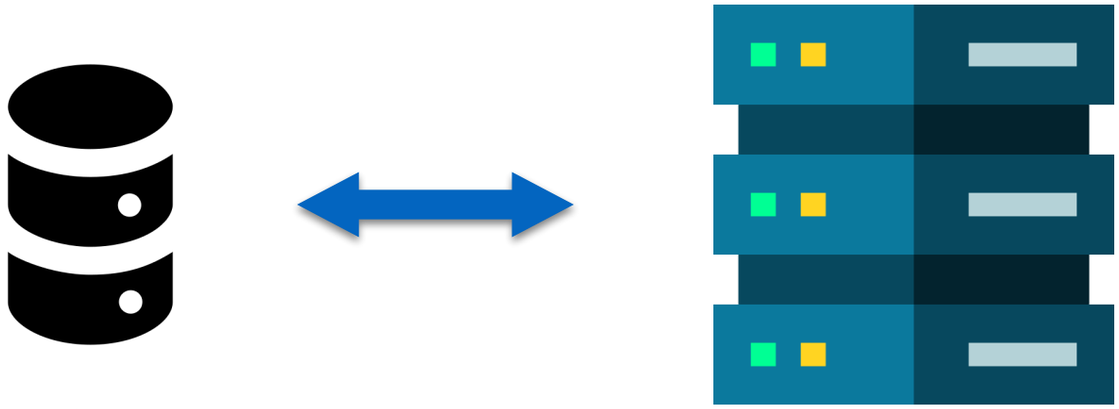
¿Cómo extraemos conocimiento que viene implícito en los datos?

- Data Mining / ML
- Statistical Learning
- IA
- Análisis de redes
- ...

Pero para sacar valor de los datos necesitamos la infraestructura adecuada

La infraestructura para poder analizar datos

Partamos por la aplicación más sencilla mencionada antes



Una base de datos de una aplicación suele ser transaccional

Una BD transaccional

Una BD transaccional está pensada para hacer muchas operaciones livianas al día:

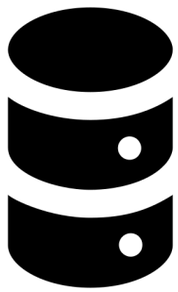
- Inserción, actualización y eliminación de tuplas
- Búsquedas simples: traer el usuario **x**, buscar los productos de la compra **y**, ...
- Además queremos soporte para propiedades ACID

Análisis de datos

Pero en general:

- Estos sistemas no están pensados para realizar consultas pesadas de análisis de datos
- No es buena idea hacer análisis directamente en la base de datos de producción

Además en el futuro tendremos más fuentes de datos



Análisis de datos

Vamos a querer hacer análisis de datos en sistemas especializados para esto: BigQuery, Redshift, Snowflake

Estos sistemas no son transaccionales, ya que están pensadas para hacer pocas consultas por día, pero cada una requiere hartos recursos computacionales

Aquí buscaremos consolidar todos los datos necesarios para los analistas

Data Lake

Un Data Lake es el lugar donde guardo todos los datos generados por mi producto:

- Datos de la base de datos del producto
- Datos del tracking de los usuarios
- Datos de nuestros experimentos (ej. AB testing)

En general, es **información cruda** sin estructura: archivos de distinto tipo, de distintos dominios, etc

Pero en algún momento vamos a procesar estos datos

Data Warehouse

Un Data Warehouse es el lugar en el que guardamos información procesada

En general, son datos puestos a disposición de los analistas para que obtengan valor de los datos

Ejemplo

Una compañía puede tener un repositorio de archivos (como S3 o Cloud Storage) con mucha información cruda; este sería el **lago de datos**

Podemos tener *jobs* que procesen estos archivos y le den estructura y sentido

La información procesada la guardamos en una base de datos de análisis (ej. BigQuery); este lugar con datos estructurados será nuestro **Data Warehouse**

Ingeniería de Datos

Ingeniería de Datos

La Ingeniería de Datos se trata de diseñar e implementar la infraestructura para trabajar con datos

Perfil ligado al desarrollo, pero con conocimiento en datos

Pensemos en las etapas de un pipeline de datos

Data Lake

Data Warehouse

Data Processing

EDA

Modelos de ML

Evaluar Modelos

ML Producción

¿Qué corresponde a ingeniería de datos vs ciencia de datos?

Ingeniería de Datos

Ciencia de Datos

Data Lake

Data Warehouse

Data Processing

EDA

Modelos de ML

Evaluar Modelos

ML Producción

¿Cómo construimos el pipeline?

Pipeline de datos

Un pipeline de datos suele usar el proceso ETL (Extract-Transform-Load) o ELT (Extract-Load-Transform)

Como el almacenamiento es bastante barato hoy en día, optamos por ELT

Requisitos

En general, para construir un pipeline de datos, se recomienda tener las siguientes habilidades:

- Programar muy bien (ojalá en un lenguaje como Python)
- Lo anterior incluye tener nociones sobre control de versiones (ej. GIT), servidores, uso de terminal, ...
- Saber sobre BDs relacionales, tanto de modelación como SQL
- Tener conocimiento sobre sistemas en la nube (ej. GCP)

¿Cómo se ve en un alto nivel un pipeline de datos?

Nuestros jobs

En un pipeline de datos vamos a tener **jobs**, que serían trozos de código que cumplen tareas específicas. Por ejemplo:

- El código que copia una fuente de datos de un lugar a otro
- El código que crea las vistas en la base de datos
- El código que entrena un modelo de Machine Learning

Orquestador

Necesitamos una herramienta que nos diga a qué hora se ejecuta cada job, y cuando un job termina diga cuál es el job siguiente

Uno podría pensar hacer esto solamente con CronJobs, pero podemos buscar una herramienta más allá de esto

Poder de Cómputo

Para ejecutar nuestros jobs necesitamos poder de cómputo

Este poder de cómputo lo podemos obtener mediante servidores on-premise, máquinas virtuales o bien en la nube mediante componentes serverless,

Stack de tecnologías

Un stack moderno para construir un pipeline de datos incluye las siguientes tecnologías:

- Airflow (u otro framework de orquestación de *jobs*)
- Docker y Kubernetes
- DBT
- Herramientas en la nube (GCS, BigQuery, Cloud Composer, GKE)

Buscando el valor de los datos

Navegando los datos

En este curso nos vamos a centrar en **las técnicas para sacar valor a los datos**

Vamos a responder la pregunta: si disponemos de **muchos datos**, ¿cómo podemos procesarlos?

En las siguientes clases

Recordar las técnicas clásicas de las bases de datos relacionales

Entender las técnicas modernas de Data Warehousing

Aprender a utilizar Google BigQuery

Procesamiento de Datos Masivos

Clase 01 - Introducción