ESPECIALIZACIÓN

Ingeniería de datos con Azure

Curso: Pipelines para Data No estructurada y Big Data

Docente: Richard Tadeo Zenteno

REGLAS

Se requiere **puntualidad** para un mejor desarrollo del curso.

Para una mayor concentración **mantener silenciado el micrófono** durante la sesión.

Las preguntas se realizarán **a través del cha**t y en caso de que lo requieran **podrán activar el micrófono**.

Realizar las actividades y/o tareas encomendadas en **los plazos determinados**.

Identificarse en la sala Zoom con el primer nombre y primer apellido.

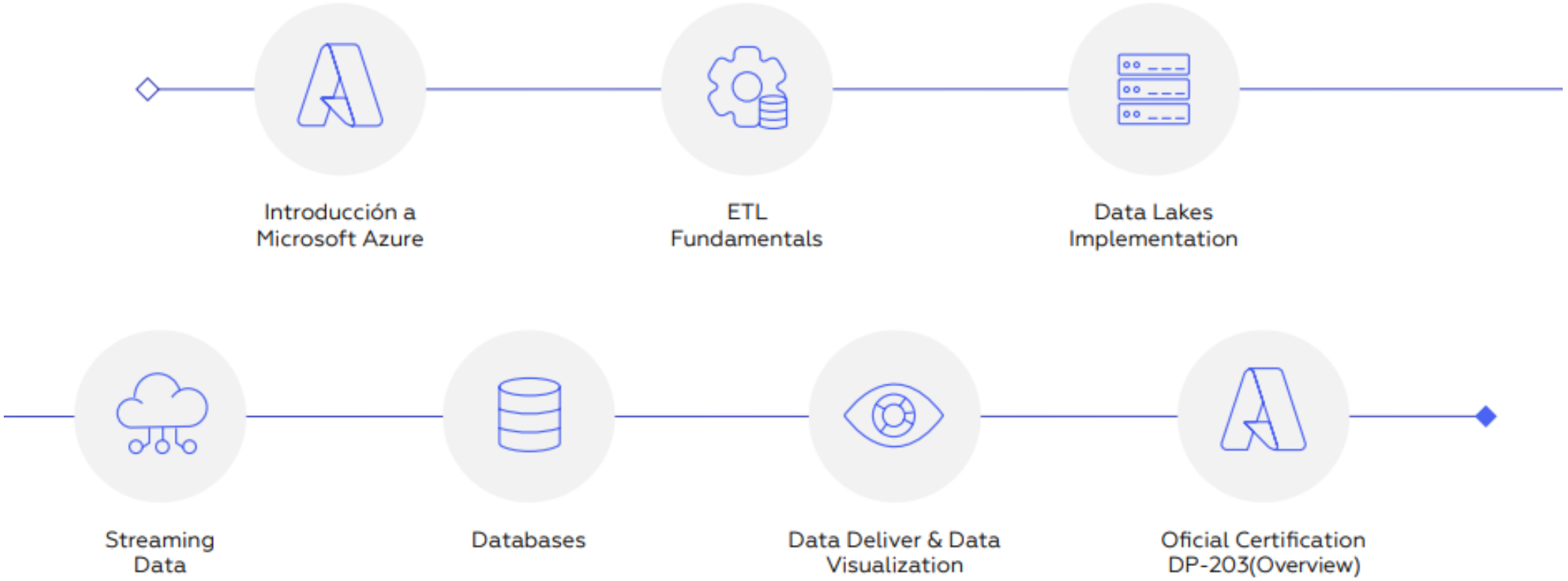
ITINERARIO

07:00 PM – 07:30 PM	***Soporte técnico DMC***
07:30 PM – 08:50 PM	***Agenda***
08:50 PM – 09:00 PM	***Pausa Activa***
09:00 PM – 10:30 PM	***Agenda***

Horario de Atención Área Académica y Soporte

Lunes a Viernes 09:00 am a 10:30 pm / Sábado 09:00 am a 02:00pm

MALLA CURRICULAR



CERIFICACIÓN FINAL
por **Aprobación** de la Especialización en **Ingeniería de Datos con Microsoft Azure** (48 horas académicas)

CONTENIDO

Introducción a Microsoft Azure

- Introducción a Cloud Computing. Proveedores de servicios Cloud, On-Premise vs. On-Cloud, principales servicios, descripción de los modelos de costos.
- Identify and Access Management (IAM). Overview de los roles principales, ejemplos de gestión de permisos.

ETL Fundamentals

- Introducción a las soluciones ETL. Definición, descripción de sus etapas.
- Introducción a los servicios Azure Data Factory y Data Flow. Características generales, casos de uso.
- Taller: Implementación de un ETL Básico con Azure.

Data Lakes Implementation

- Introducción a Data Lakes. Definición, arquitectura, capas (Raw, Stage, Analytics).
- Introducción a los servicios Azure Blob Storage y Storage Account.
- Taller: Implementación de un Datalake en Azure.

CONTENIDO

Streaming Data

- Introducción a procesamiento de datos Batch y Streaming. Diferencias Near-Real-Time y Real-Time.
- Introducción a IoT. Definición, uso de sensores, aplicaciones.
- Revisión de servicios: Azure EventHubs y IoTHub. Características generales, ejemplos de implementación y uso.
- Taller: Manejo de Streaming al Data.

Databases

- Introducción a las bases de datos Relacionales y No-Relacionales. Definición, características, casos de uso.
- Azure SQL Database for MariaDB. Descripción y características generales.
- Azure SQL Database for PostgreSQL. Descripción y características generales.
- Azure SQL Database for CosmosDB. Descripción y características generales.
- Taller: Diseño de una base de datos relacional y técnicas para poblarla.

Data Deliver & Data Visualization

- Azure Synapse Analytics. Propósito del servicio, características generales.
- Fabric. Propósito del servicio, características generales.
- Taller: Conexión de Power BI a servicios de datos de Azure.

AGENDA

01

Introducción a procesamiento de datos Batch y Streaming

02

Revisión de servicios: Azure EventHubs y IoTHub

03

Introducción a IoT

04

Taller: Manejo de Streaming al Data

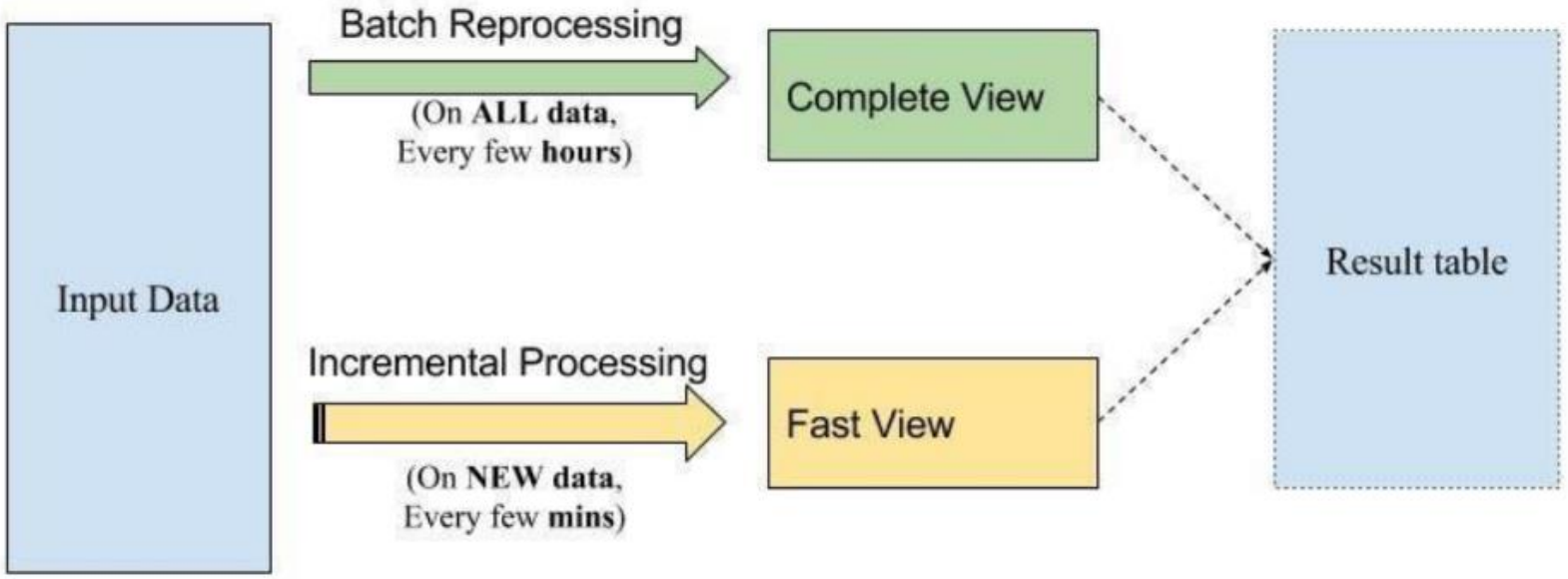
Métodos de Procesamiento

- **Batch:** La ingesta de datos en batch implica recopilar y procesar datos en grupos o lotes grandes en intervalos de tiempo específicos. Este método es adecuado para datos que no requieren procesamiento en tiempo real. Es usado generalmente para Análisis diario de ventas, procesamiento nocturno de datos acumulados durante el día, actualizaciones periódicas de bases de datos.

- **Streaming:** La ingesta de datos en streaming se refiere a la recopilación y procesamiento de datos en tiempo real a medida que se generan. Este método permite el análisis y la toma de decisiones inmediatas basadas en la información más actualizada. Es usado generalmente para Monitoreo en tiempo real de redes sociales, análisis de transacciones financieras instantáneas, sistemas de alerta temprana.

- **Incrementales:** La ingesta de datos incrementales implica procesar solo los nuevos datos o los que han cambiado desde la última carga, en lugar de procesar todo el conjunto de datos. Esto mejora la eficiencia al reducir el volumen de datos procesados. Es usado generalmente para Sincronización de bases de datos, actualizaciones de catálogos de productos en e-commerce, refrescos periódicos de data warehouses.

Métodos de Procesamiento



Formatos de archivos de procesamiento

APACHE PARQUET

Es un formato de almacenamiento columnar disponible para cualquier proyecto en el ecosistema de Hadoop, independiente del framework utilizado para procesar los datos o el lenguaje de programación.

Parquet fue creado para que las ventajas de la compresión y la eficiencia de la representación columnar estuviera disponible para cualquier proyecto del ecosistema Hadoop.

Parquet está diseñado para soportar eficientemente esquemas de compresión y codificación.
Parquet permite que los esquemas de compresión se especifiquen en un nivel por columna. Parquet tiene tres opciones para la compresión de sus archivos: **snappy, gzip o ninguno**.

Formatos de archivos de procesamiento

APACHE AVRO

Es un sistema de compresión diseñado por Apache para el proyecto Hadoop para la serialización de datos que proporciona:

- Estructuras de datos complejas
- Un formato binario, compacto y rápido
- Un fichero contenedor, para guardar datos persistentes

Avro se basa en **esquemas**. Cuando los datos *.avro* son leídos siempre esta presente el esquema con el que han sido escritos. Esto permite aumentar el rendimiento al escribir los datos, haciendo la serialización rápida y viable en espacio. También facilita el uso de lenguajes dinámicos de scripting porque los datos se encuentran con su esquema, lo que les hace auto-descriptivos.

Los esquemas de Avro se definen mediante JSON para facilitar la implementación en otros lenguajes que ya disponen de librerías JSON.

```
{
 "namespace":"example.avro",
 "type":"record",
 "name":"User",
 "fields":[
  {
   "name":"name",
   "type":"string"
  },
  {
   "name":"favorite_number",
   "type":[
    "int",
    "null"
   ]
  },
  {
   "name":"favorite_color",
   "type":[
    "string",
    "null"
   ]
  }
 ]
}
```

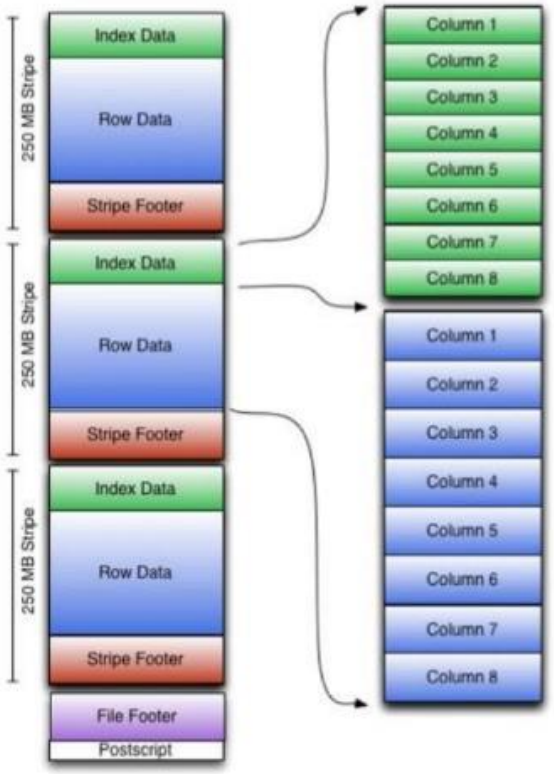
Formatos de archivos de procesamiento

APACHE ORC

En Enero de 2013 Apache creó los archivos ORC (Optimized Row Columnar) como parte de una iniciativa para incrementar la velocidad de Hive y mejorar el almacenamiento de los datos en HDFS. EL objetivo era **alta velocidad de procesamiento en un reducido tamaño de archivo**.

ORC es un **formato columnar autodescriptivo y "sin tipo",** diseñado para carga de trabajos en Hadoop. Tiene la ventaja de los formatos columnares que permiten al *Reader* leer, descomprimir y procesar solo los valores que necesite para la consulta actual. Que es un formato "sin tipo" significa que el *Writer* será el encargado de escoger la codificación más adecuada y generar un índice interno a medida que se escribe el archivo.

Un archivo ORC contiene grupos de **filas llamados *Stripes***, junto con la información auxiliar al final del archivo (*Stripe Footer*). El **final del archivo es un *Postscript* que contiene los parámetros de compresión y el tamaño del *Stripe Footer*.**



Formatos de archivos big data

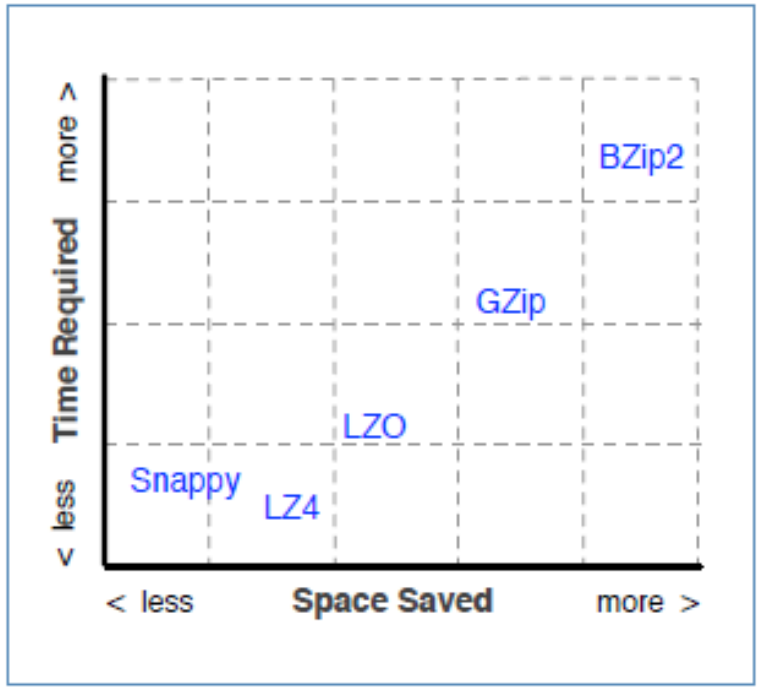
Properties	CSV	JSON	Parquet	Avro
Columnar	✗	✗	✓	✗
Compressable	✓	✓	✓	✓
Splittable	✓*	✓*	✓	✓
Readable	✓	✓	✗	✗
Complex data structure	✗	✓	✓	✓
Schema evolution	✗	✗	✓	✓

@luminousmen.com

https://luminousmen.com/post/big-data-file-formats

Códec Compresión

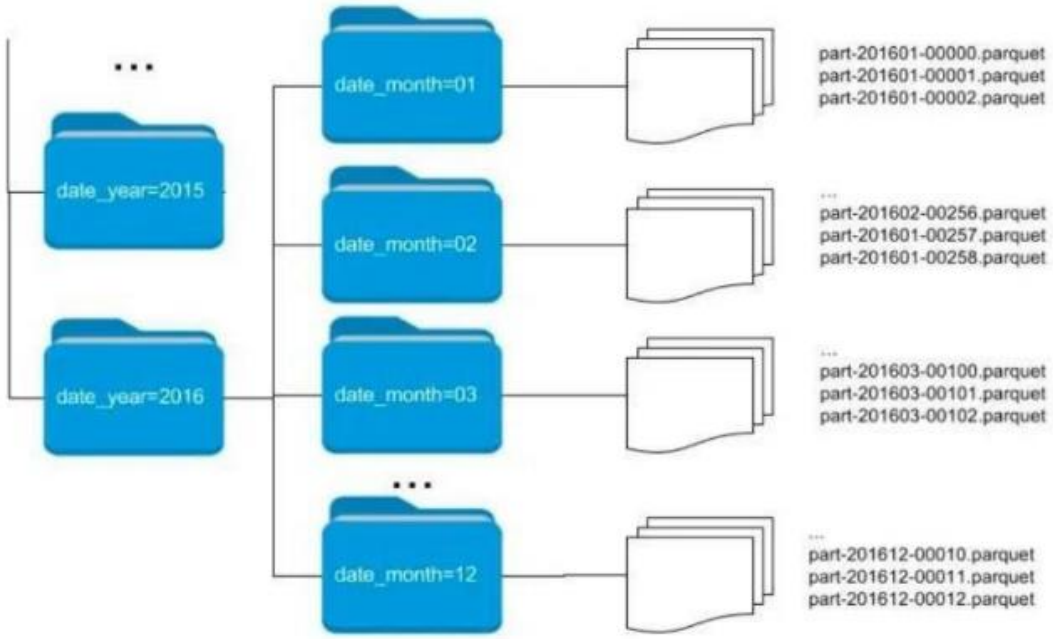
Codec	Best Used For	File Type
GZip	• "Cold" data (infrequently accessed)	• Text • Parquet • Avro • RCFile • SequenceFile
BZip2	• "Cold" data (infrequently accessed)	• Text • Avro • RCFile • SequenceFile
LZO	• "Hot" data (frequently accessed)	• Text • RCFile • SequenceFile
Snappy	• "Hot" data (frequently accessed)	• Text • Avro • Parquet • RCFile • SequenceFile



Organización y Particionamiento de Datos en un Datalake

- **Organización de Datos**: Se refiere a cómo se estructuran los datos dentro del Data Lake. **Esto puede incluir la creación de zonas o capas (como raw, curated, y consumable)** para separar los datos según su grado de procesamiento y propósito.

- **Particionamiento de Datos**: Implica **dividir los datos en particiones o segmentos más pequeños** basados en ciertos criterios (como fecha, región geográfica, o categoría de producto). Esto mejora el rendimiento de las consultas al limitar la cantidad de datos que necesitan ser escaneados.



Laboratório: Manejo de Particiones en Databricks

RONDAS DE PREGUNTAS

¡GRACIAS!