





Fabric
User Group
ESP



Fabric
User Group
ESP



Fabric Christmas Presents

AGENDA DE SESIONES



26 de diciembre – Cómo utilizo GitHub Copilot con el formato **PBIP** para trabajar más rápido – Toni Ferrá



27 de diciembre – ¡El fallar se va a acabar! Asegura el éxito de tus despliegues CI/CD en Fabric y Power BI – Rafael Báguena



28 de diciembre – Seguridad en Almacenes de Datos en Microsoft Fabric – Germán Cayo



29 de diciembre – Transforma tus datos: de SAP B1 a Dashboards Inteligentes con Fabric – Raúl Sao



30 de diciembre – Notebooks en Fabric: ¿por dónde empiezo sin entrar en pánico? – Mónica Mesa



La Navidad nos trae los primeros regalos de Microsoft Fabric 🎁

Estas fiestas arrancamos con una semana muy especial para la comunidad de datos: **5 charlas técnicas, 5 días consecutivos y formato 100 % online, del 26 al 30 de diciembre.**

Empecemos por el principio



Mónica Mesa

Data & Analytics Engineer

Especializada en Power BI y Microsoft Fabric

Ayudo a convertir datos en decisiones reales: desde arquitecturas analíticas y modelos bien gobernados hasta soluciones que se usan, escalan y aportan valor.

Me muevo entre Power BI, Microsoft Fabric, Spark, SQL y PySpark, con especial interés en cómo la analítica avanzada y la automatización para hacer más sencillo lo que antes parecía complicado.

Soy **Microsoft MVP en Data Platform** y participo activamente en la comunidad: co-organizadora de **Women in Fabric** y del **Fabric User Group ESP**, y colaboradora en eventos como **Power BI Days**, **W4TT** y otras iniciativas de datos.

Me apasiona compartir conocimiento, enseñar desde la práctica y demostrar que incluso lo más técnico se puede entender... y disfrutar.

#FabricEvangelizator #datalover

www.monicamesa.es

www.calendarioeventosdatos.com

<https://www.linkedin.com/in/monicamesa>



Notebooks en Fabric: ¿por dónde empiezo sin entrar en pánico?



Para quienes vienen de Power BI y no quieren sufrir en el intento

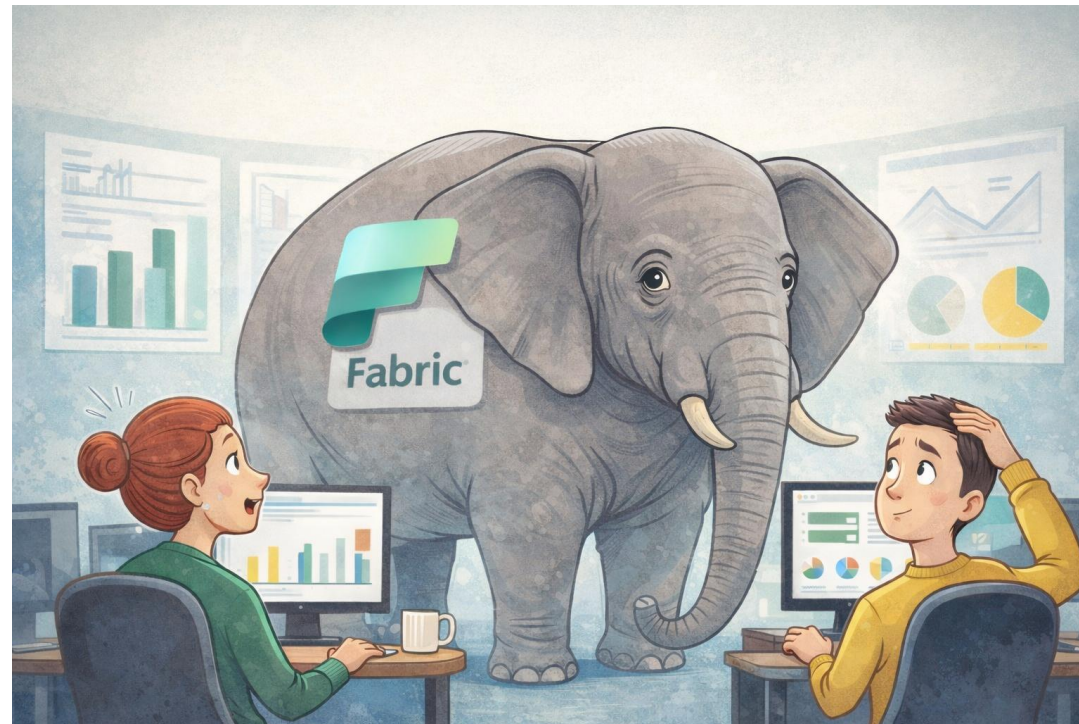
Agenda

1. Qué es realmente un notebook
2. Anatomía de un notebook
3. Las 3 piezas clave: código – motor - lenguaje
4. Ejecución
5. Creando mi primer notebook
6. Tareas básicas
7. SQL en Spark
8. Funciones UDF
9. Regalo final

El elefante en la habitación

Fabric impone. Digámoslo.

- Nuevos conceptos
- Nuevas pantallas
- Nuevos nombres
- Nuevas dudas



“Yo venía a hacer informes... ¿por qué ahora tengo Spark delante?”

Fabric ≠ solo dataflows

1

Carga de datos “clásica”

Carga manual
Dataflows Gen 2
Pipelines (Copy Data)
Copy Job

2

Carga sin copiar datos

Shortcuts
Mirroring

3

Procesamiento y control

Notebooks (Cuadernos)

4

Tiempo Real (RTI)

Eventstreams

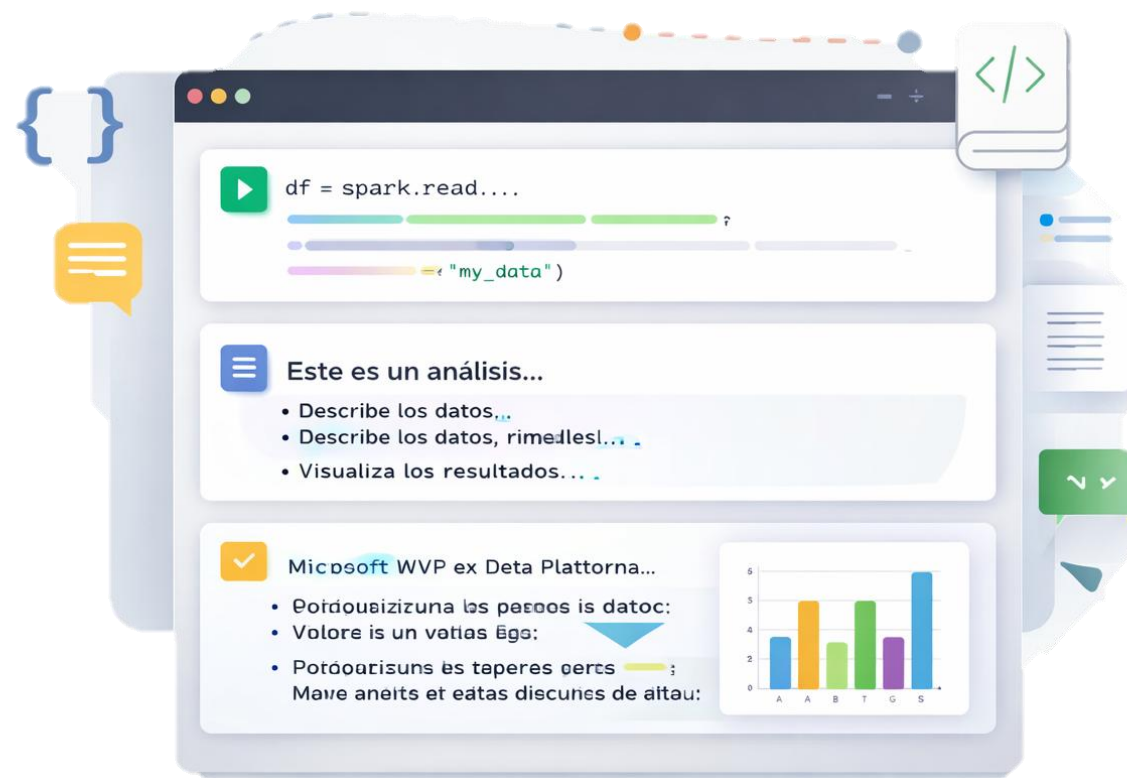
Entonces... ¿qué pinta un notebook aquí?



Vale, pero... ¿qué es realmente un notebook?

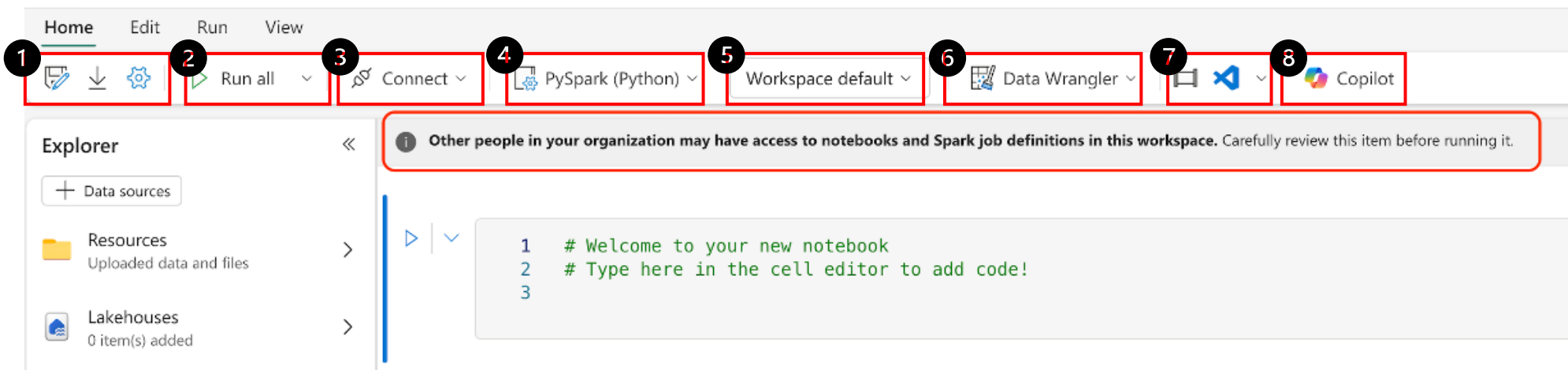
Un notebook es un **bloc de notas interactivo** donde mezclas:

- código
- texto
- resultados



Nada más. Nada menos.

Anatomía de un notebook (1/2)



Aquí están los controles principales del menú Home (barra superior):

1. Guardar como, descargar, configuración

Acciones básicas para gestionar el notebook

2. Run / Run all

El botón del poder: ejecutas una celda...
o todo el notebook sin miedo (o con él).

3. Connect

Conecta el notebook a un motor de ejecución (Spark o SQL Engine).

4. Lenguaje del notebook (por ejemplo: *PySpark (Python)*)

Define el lenguaje por defecto de las nuevas celdas.

5. Enviroment

El “contexto técnico” del notebook:
versiones de librerías, dependencias, configuraciones..

6. Data Wrangler

Herramienta visual para explorar y transformar datos
sin necesidad de saber escribir código.

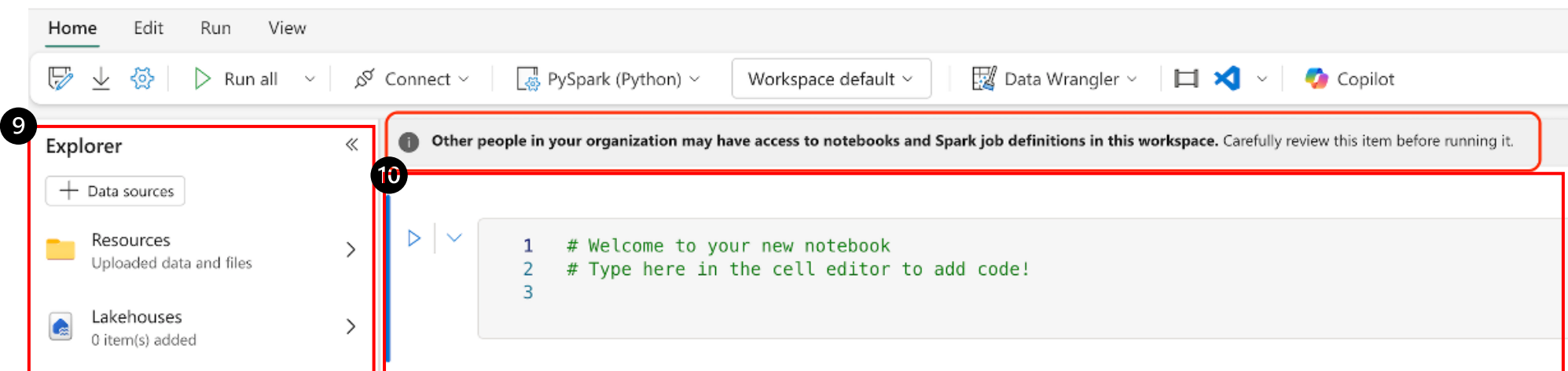
7. Agregar a pipeline y abrir en VSC

Para automatizar y llevar el notebook a producción.

8. Copilot

Tu copiloto de código. No pilota por ti, pero ayuda bastante.

Anatomía de un notebook (2/2)



9. Explorer

Desde aquí accedes a:

- Data sources
- Lakehouses
- Tablas
- Archivos
- Recursos del notebook

10. Celdas (canvas central)

El corazón del notebook:

- Celdas de **código**
- Celdas de **texto (Markdown)**
- Resultados visibles justo debajo

Código · Motor · Lenguaje

En un notebook hay tres elementos clave distintos que trabajan en conjunto



El motor

Un notebook necesita un motor que ejecute el código

El motor es quien hace **el trabajo duro**

En Fabric hay más de un motor:

Notebooks de Spark
→ Motor Apache Spark



Diseñado para el procesamiento de grandes volúmenes de datos.

Notebooks de T-SQL
→ Motor SQL Engine



Optimizado para realizar consultas analíticas.

¡Empieza sin configurar nada!

Fabric te da una configuración por defecto que funciona para la mayoría de los casos. La configuración avanzada no es obligatoria.

Apache Spark: el “musculman” del notebook

- Motor de procesamiento distribuido (en varias máquinas)
- Diseñado para trabajar con muchos datos y hacerlo rápido
- Usado en ingeniería y ciencia de datos
- Permite paralelizar trabajo y escalar sin cambiar tu forma de pensar
- En Fabric, Spark está integrado y listo para usar



Qué es un clúster (el “cómo se ejecuta”)



1. TÚ (El Cerebro)

Escribes el código para definir qué hacer con los datos.

2. SPARK (El Capataz)

Recibe tu código y reparte el trabajo para procesarlo de forma masiva y rápida.

3. FABRIC (El Maquinista)

Proporciona las máquinas (el clúster) que Spark necesita, encendiéndolas y apagándolas automáticamente.

? ¿Y QUÉ ES UN CLÚSTER?



Un equipo de máquinas trabajando juntas.

Es la potencia de cálculo que Spark utiliza para ejecutar tus instrucciones.



Fabric los gestiona por ti.

No tienes que crearlos, mantenerlos ni configurarlos. ¡Se activan y desactivan solos!



Para empezar: usa los “Starter Pools”.

Son clústeres preconfigurados y listos para usar al instante. ¡Sin complicaciones!

Celdas: pequeñas piezas, de un gran puzzle

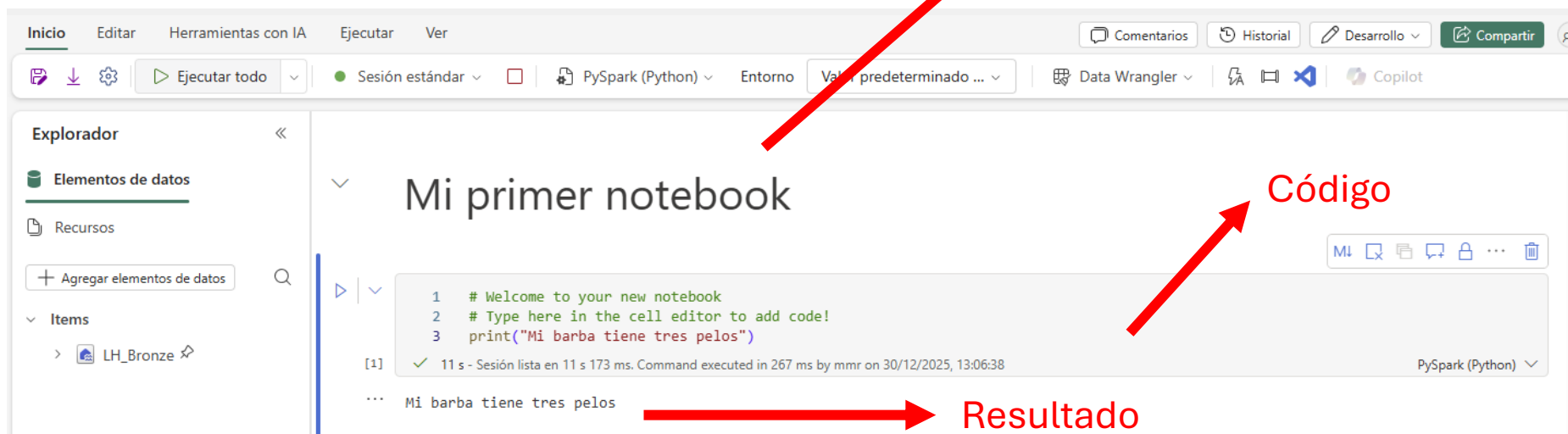
Un notebook se divide en **celdas** y cada celda representa una **acción concreta** (que no croqueta)

Hay celdas de:

- **Código**
- **Texto (Markdown)**

Puedes ejecutar **una celda o varias**

Ves el resultado **al momento**



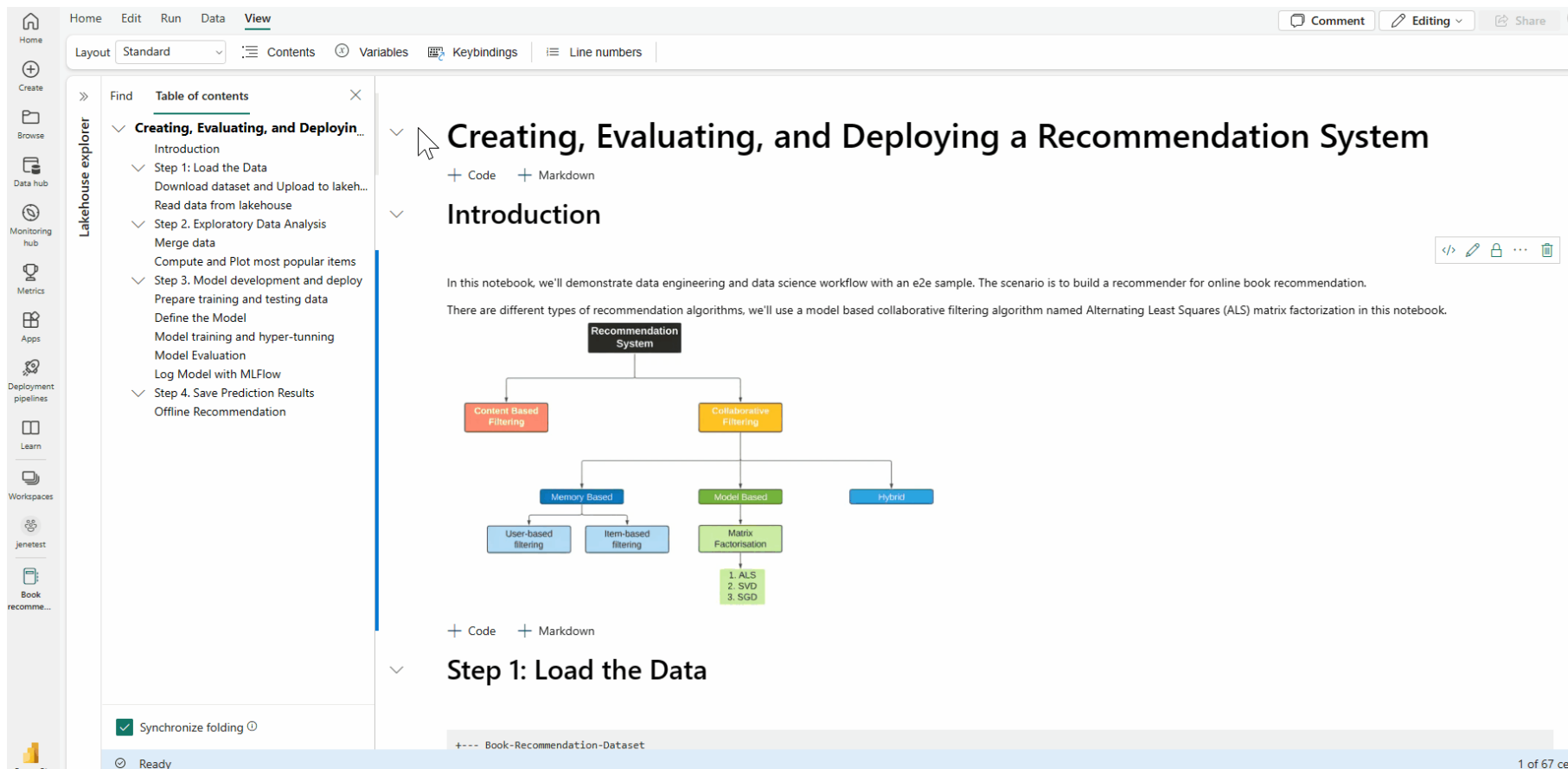
The screenshot shows a Jupyter notebook titled "Mi primer notebook". The interface includes a top menu bar with options like "Inicio", "Editar", "Herramientas con IA", "Ejecutar", and "Ver". Below the menu is a toolbar with icons for saving, downloading, settings, and running code. The left sidebar shows the "Explorador" (Explorer) panel with "Elementos de datos" and "Recursos". The main area displays a code cell with the following content:

```
1 # Welcome to your new notebook
2 # Type here in the cell editor to add code!
3 print("Mi barba tiene tres pelos")
```

Below the code cell, the output is shown: "[1] ✓ 11 s - Sesión lista en 11 s 173 ms. Command executed in 267 ms by mmr on 30/12/2025, 13:06:38". The output text "Mi barba tiene tres pelos" is displayed below the execution bar. Red arrows point from the text labels to the corresponding parts of the notebook: "Markdown" points to the top of the code cell, "Código" points to the code text, and "Resultado" points to the output text.

Celdas: pequeñas piezas, de un gran puzzle

Ejemplo de Markdown Folding



The screenshot shows a Databricks notebook interface. On the left, a 'Lakehouse explorer' sidebar displays a 'Table of contents' for the notebook 'Creating, Evaluating, and Deploying a Recommendation System'. The notebook content is visible on the right, showing the title, introduction, and a diagram of recommendation algorithms. The diagram illustrates the structure of a Recommendation System, branching into Content Based Filtering and Collaborative Filtering. Collaborative Filtering further branches into Memory Based (User-based filtering, Item-based filtering) and Model Based (Matrix Factorisation, which includes ALS, SVD, and SGD). A 'Hybrid' option is also shown. The notebook is currently on 'Step 1: Load the Data'.

Table of contents:

- Creating, Evaluating, and Deploying a Recommendation System
 - Introduction
 - Step 1: Load the Data
 - Download dataset and Upload to lakeh...
 - Read data from lakehouse
 - Step 2: Exploratory Data Analysis
 - Merge data
 - Compute and Plot most popular items
 - Step 3: Model development and deploy
 - Prepare training and testing data
 - Define the Model
 - Model training and hyper-tuning
 - Model Evaluation
 - Log Model with MLFlow
 - Step 4: Save Prediction Results
 - Offline Recommendation

Creating, Evaluating, and Deploying a Recommendation System

+ Code + Markdown

Introduction

In this notebook, we'll demonstrate data engineering and data science workflow with an e2e sample. The scenario is to build a recommender for online book recommendation.

There are different types of recommendation algorithms, we'll use a model based collaborative filtering algorithm named Alternating Least Squares (ALS) matrix factorization in this notebook.

```

graph TD
    RS[Recommendation System] --> CBF[Content Based Filtering]
    RS --> CF[Collaborative Filtering]
    CF --> MB[Memory Based]
    CF --> ModelBased[Model Based]
    CF --> Hybrid[Hybrid]
    MB --> UBF[User-based filtering]
    MB --> IBF[Item-based filtering]
    ModelBased --> MF[Matrix Factorisation]
    MF --> List["1. ALS<br/>2. SVD<br/>3. SGD"]
  
```

+ Code + Markdown

Step 1: Load the Data

--- Book-Recommendation-Dataset

1 of 67 cells

Los lenguajes (idiomas querida)

En un notebook puedes trabajar con varios lenguajes:



Puedes cambiar de lenguaje:

- con **magic commands** (%%sql, %%pyspark, etc.)
- o desde la **esquina inferior derecha de la celda**

La ejecución



Ejecución Interactiva

El Explorador
Roboto Medium

Control total para crear y depurar.



Ejecución por Canalización

El Orquestador
Roboto Medium

Integración en flujos de trabajo complejos.



Ejecución Programada

El Autómata
Roboto Medium

Automatización desatendida y recurrente.



El Explorador: Ejecución Interactiva

El control absoluto en tus manos.

Disparador: Manual, por un usuario.

Mecanismo:

Directamente desde la interfaz del notebook o vía API.

Se ejecuta con la identidad de: **Tú mismo.**

El notebook hereda todos tus permisos de acceso a datos.

Caso de uso ideal:

Desarrollo de código, exploración de datos, depuración y pruebas unitarias.



El Orquestador: Ejecución por Canalización

Tu notebook como parte de un plan mayor.

Disparador: Una actividad dentro de una canalización (pipeline) de Fabric.

Mecanismo: Se invoca como un paso dentro de un flujo orquestado.

Se ejecuta con la identidad de: El **último usuario que editó y guardó** la canalización.

Caso de uso ideal: Procesos ETL/ELT, ejecución condicionada por otras tareas, flujos de producción.



El Autómata: Ejecución Programada

Fiabilidad y puntualidad sin intervención humana.

- **Disparador:** Un horario predefinido (ej. diario, semanal).
- **Mecanismo:** Configuración de la programación directamente en las propiedades del notebook.
- **Se ejecuta con la identidad de:** Quien **creó o modificó por última vez** la programación.
- **Caso de uso ideal:** Actualizaciones de datos recurrentes, generación de informes periódicos, tareas de mantenimiento.



Me has convencido, voy a crear uno

Tu aventura empieza con una pregunta.

Olvida la tecnología por un momento. Simplemente pregúntate:

¿Cuál es mi tarea ahora mismo?

¿Necesito procesar y transformar datos brutos para dejarlos listos?



Crea un Notebook de Spark desde un Lakehouse o desde “Nuevo Elemento” y usa PySpark, Scala, SparkSQL, R..

¿Necesito consultar y analizar datos que ya están limipos en mi Warehouse?



Crea un Notebook de T-SQL desde un Warehouse o un SQL-endpoint y lanza tus queries de inmediato

DEMO TIME: Vamos a verlo?



Cositas básicas

El patrón que se repite siempre

- Ingestar datos
- Explorar qué ha llegado
- Entender estructura y tipos
- Transformar información
- Preparar datos para análisis

Operaciones habituales

- Agrupar y ordenar datos
- Relacionar datos (joins entre dataframes)
- Guardar resultados en formato Delta

Y si necesitas el comodín del público

Go Go Power Wranglers!!!

Data Wrangler



Le podéis preguntar a Rafa Báguena o a Raquel García que dieron una sesión buenísima en el Power BI Days de Santiago de Compostela (2024) y aún la recuerdo.



Rafael Báguena Girbés · 1er

Strong alone, stronger together | Power BI | Microsoft Fabric | Azure
Valencia/València, Comunidad Valenciana / Comunitat Valenciana, España ·

<https://www.linkedin.com/in/rafaelbaguena/>
<https://github.com/l2aFa/power-wranglers>



Raquel García Jaén · 2º

Data Scientist | AI Engineer

Lugo, Galicia / Galiza, España · [Información de contacto](#)

<https://www.linkedin.com/in/raquel-garcía-jaén-b645a6188/>

Tu conocimiento de SQL es tu mejor aliado en Spark.

No dejes que nombres como 'Spark' o 'PySpark' te intimiden. Puedes empezar a trabajar con el motor más potente de Fabric usando una sintaxis que ya te resulta familiar: **SparkSQL**.

Misma lógica, diferente dialecto.

En T-SQL dices TOP 10



en SparkSQL dices LIMIT 10

En T-SQL usas ISNULL()



en SparkSQL usas COALESCE()

En T-SQL usas CONVERT()



en SparkSQL usas CAST()

En T-SQL usas GETDATE()



en SparkSQL usas current_timestamp()



La estructura que ya dominas (SELECT, FROM, WHERE, GROUP BY) es prácticamente idéntica. Empieza por aquí y gana confianza.

La chuleta: T-SQL, Spark SQL y PySpark.

La misma idea, tres formas de escribirla.

Qué quiero hacer	T-SQL	Spark SQL	PySpark
Leer datos	SELECT * FROM tabla	SELECT * FROM tabla	spark.read.table('tabla')
Filtrar filas	WHERE col = 'A'	WHERE col = 'A'	df.filter(col('col') = 'A')
Seleccionar columnas	SELECT col1, col2	SELECT col1, col2	df.select('col1','col2')
Agrupar	GROUP BY col	GROUP BY col	df.groupBy('col')
Agregar	SUM(col)	SUM(col)	df.agg(sum('col'))
Ordenar	ORDER BY col	ORDER BY col	df.orderBy('col')
Guardar resultado	CREATE TABLE AS	CREATE TABLE AS	df.write.saveAsTable()

Guarda esta diapositiva. Es tu mapa para moverte entre los diferentes lenguajes dentro de Fabric.



Nivel pro

Cuando aprendas y te sientas con confianza, tienes más herramientas:

- Parámetros de notebook
- Reutilizar notebooks con %run
- Configurar sesiones Spark (%%configure)
- Copilot para ayudarte a escribir código
- Widgets para interacción

El momento “ahora lo entiendo”

**No todo son dataflows.
Y eso es una buena noticia.**

- Hay más opciones
- Más control
- Más posibilidades



**Fabric impone hasta que lo abres.
El notebook es el primer paso**

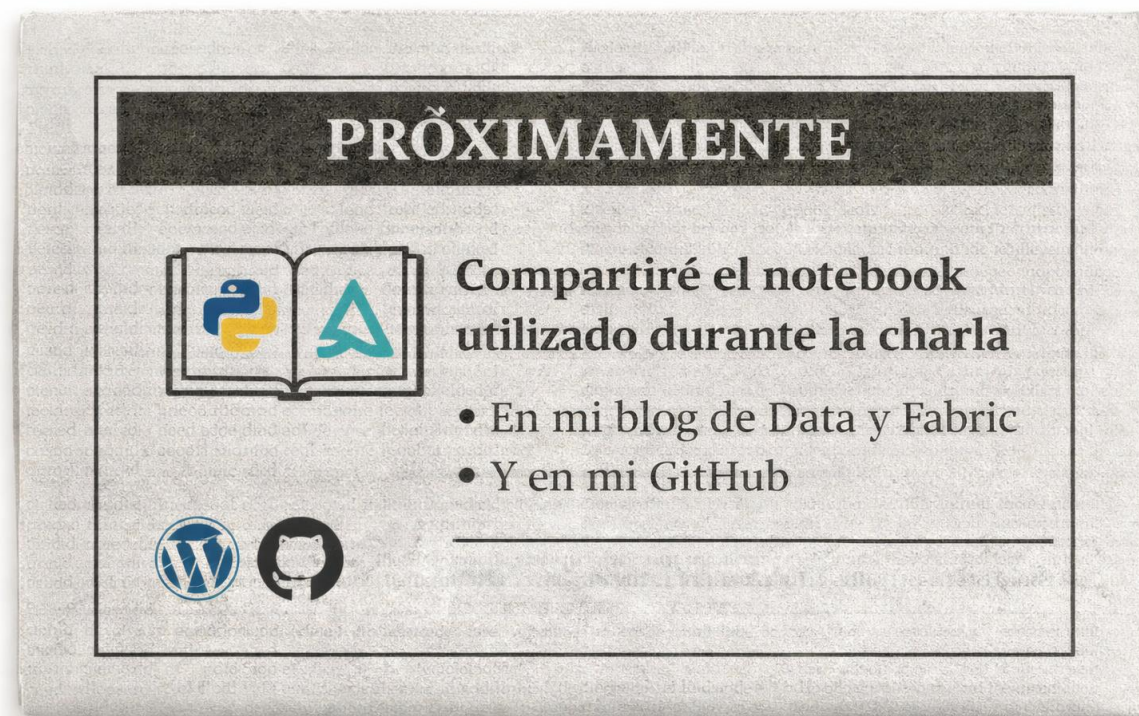


Patrocinado por:
Sara Lozano

Enlaces de interés

- Cómo usar los cuadernos de Microsoft Fabric – Microsoft Learn: <https://learn.microsoft.com/es-es/fabric/data-engineering/how-to-use-notebook>
- Let's look at using some PySpark in your first Microsoft Fabric Notebook – Guy in a Cube: <https://www.youtube.com/watch?v=K2Wdldfplic>
- Apache Spark for SQL Developers: Master DataFrames, Joins, and SQL in Microsoft Fabric – Devon and Cornwall User Group: <https://www.youtube.com/watch?v=8v-Vrm8Mh8I>
- AND Fabric: <https://www.adnfabric.com/>

Y si te quedaste hasta el final..

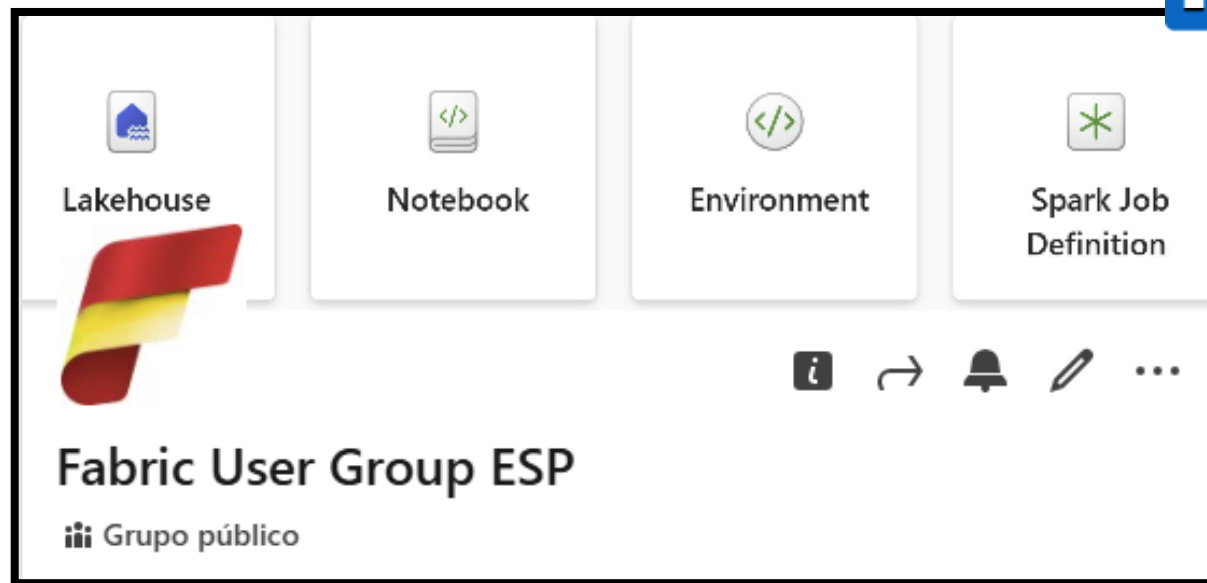


#StayTuned
#FabricIsTheWay

🎁 Fabric Christmas Presents 🎁

Síguenos en LinkedIn

<https://www.linkedin.com/groups/13099547/>



GRACIAS

This is
my
thank you
dance!

