

# Trabajo Práctico

## Base de Datos 2

### Decisiones tomadas

Para el desarrollo de este proyecto, seleccionamos dos bases de datos NoSQL: MongoDB y Neo4j. Todas las consultas fueron implementadas en Python utilizando las bibliotecas **pymongo** y **neo4j**, lo que nos permitió centralizar y estandarizar el código.

MongoDB fue elegido para almacenar las colecciones relacionadas con **clientes** y **teléfonos** debido a su esquema flexible, que permite que cada documento JSON (equivalente a una fila en un RDBMS) tenga atributos diferentes. Esta característica es especialmente útil para la colección de teléfonos, ya que los clientes pueden tener un número variable de ellos. Además, este enfoque nos permite agregar o eliminar atributos en el futuro sin necesidad de modificar el esquema general. Dado que estos datos no requieren disponibilidad constante, MongoDB, como base de datos clasificada como **CP** según el teorema CAP, resulta adecuada para satisfacer estas necesidades. Asimismo, el lenguaje de consultas de MongoDB (MQL) es intuitivo y facilita la interacción con los datos de manera eficiente.

Para gestionar los datos relacionados con **productos** y **facturas**, optamos por Neo4j, ya que las relaciones que forman pueden representarse de manera natural en un modelo de grafos. Debido a que la facturación depende del inventario de productos, es fundamental mantener la consistencia de los nodos y las relaciones asociadas. Neo4j, al ser una base de datos **ACID**, garantiza la consistencia transaccional, lo cual es crucial para evitar errores al procesar facturas y actualizar el inventario en tiempo real. Además, el lenguaje Cypher de Neo4j facilita la escritura de consultas claras y concisas para modelar y gestionar relaciones complejas, como rastrear el historial de compras de un cliente o identificar los productos vinculados a una factura específica. La elección de Neo4j también aporta ventajas adicionales en términos de visualización de datos, ya que permite comprender de manera más intuitiva la estructura y las conexiones entre ellos.

Para la ejecución de las consultas, se decidió usar los drivers que dispone Python para poder comunicarse con las bases de datos debido a la versatilidad que dispone el lenguaje a la hora de manejar datos. La creación del índice para los productos, la población y consultas de las bases de datos disponen de su propio script de Python que pueden ser ejecutadas a través de una terminal con Python3. Las consultas que implican escrituras también reciben parámetros de entrada por consola.