HCL (BIG DATA)

TAREA I UD4

ALUMNO CESUR

25/26

Alejandro Muñoz de la Sierra

PROFESOR

Óscar González Núñez

I N T R O D U C C I O N

El sector energético cambia muy rápido. La meta ya no es solo transportar electricidad de un punto a otro. Debemos transformar la red tradicional en una **Smart Grid** o red inteligente. La empresa solicitó este proyecto. El desafío es evidente. Los datos de los contadores son tan valiosos como la propia energía.

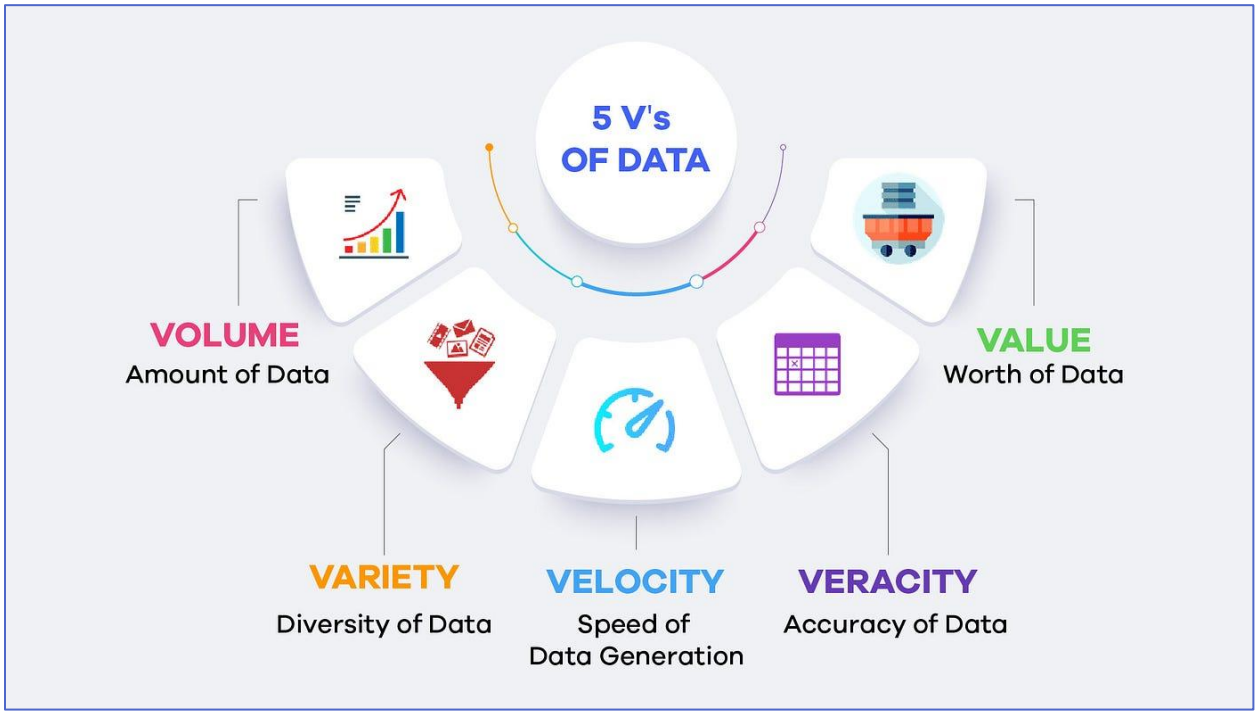
Esta propuesta técnica diseña una plataforma Big Data para procesar esa información. No queremos solo acumular datos. Buscamos usos prácticos. Entenderemos el consumo de los usuarios mediante contadores inteligentes (**Smart Meters).** La compañía abandonará las tarifas planas antiguas. Ofrecerá productos personalizados y detectará fraudes o averías antes.

0 2

DEFINICIÓN DEL PROBLEMA: ¿POR QUÉ FALLA EL SISTEMA ACTUAL?

Analizamos la petición de la compañía. Enfrentamos un problema clásico de "Ingeniería de Datos a Gran Escala". El sistema actual usa bases de datos relacionales (**SQL**). Estas bases son **insuficientes**. No soportan la carga de trabajo.

Desglosamos el problema según la teoría de las "**5V**" del **Big Data** y la aplicamos al proyecto:



Volumen: Millones de clientes envían lecturas cada 15 minutos. Esto genera Terabytes de datos al mes. Los servidores tradicionales fallan al escalar horizontalmente.

Velocidad: Conocer un pico de tensión ayer es inútil. La ingesta de datos debe ocurrir casi en tiempo real (Near Real-Time). Así reaccionaremos al instante.

Variedad: Este punto es complejo. Cruzaremos datos ordenados de lecturas con otros desordenados. Estos incluyen registros de incidencias, clima, ubicación y tipo de vivienda.

Veracidad: Los sensores IoT fallan. El sistema debe filtrar el "ruido" o lecturas erróneas. Así no afectarán a las estadísticas.

Valor: La meta final es la microsegmentación. No trataremos igual a una fábrica que a una familia de cuatro personas.

0 3
M E T O D O L O G Í A D E G E S T I Ó N

3.1. Marco Global (PMI)

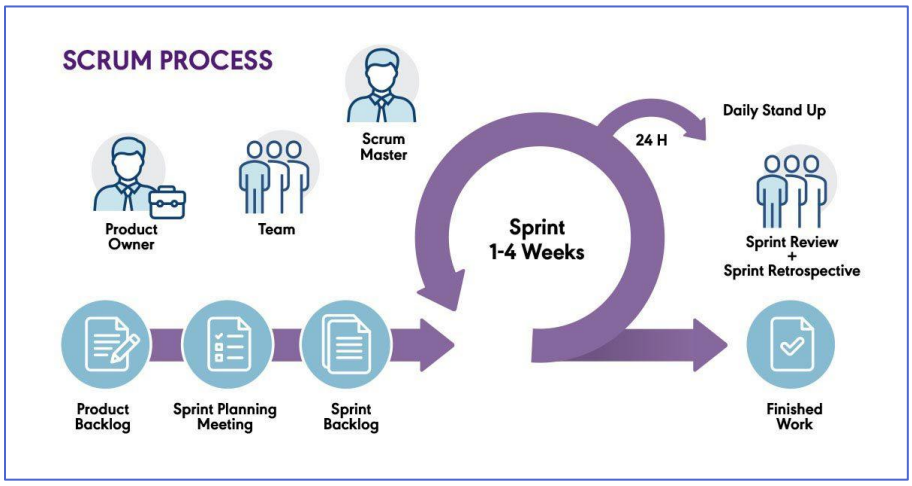
Usaremos el enfoque **PMI** para la gobernanza general del proyecto.

Alcance: Definiremos qué incluye el trabajo. El proyecto abarca la plataforma y el algoritmo de segmentación. Excluimos la aplicación móvil y la facturación. Esto evita el crecimiento descontrolado del alcance (scope creep).
Costes: Controlaremos el gasto en la nube (AWS/Azure). El presupuesto no debe perderse en licencias.

3.2. Desarrollo del Software (Scrum)

Desconocemos los patrones de consumo iniciales. Un modelo en cascada conlleva riesgos. Usaremos Scrum:

Sprints: Ciclos de tres semanas.
Roles: Habrá un Product Owner de la eléctrica experto en el negocio. Nosotros seremos el equipo de desarrollo, incluyendo Ingenieros de Datos y Arquitectos.
Backlog: Gestionaremos las tareas pendientes. usaremos historias de usuario en Jira. Priorizaremos las tareas más útiles al momento (ejemplo: "Ver consumo por código postal").



0 4
A N Á L I S I S D E R I E S G O S

Los riesgos técnicos preocupan en los proyectos de datos. Los riesgos legales asustan más. Preparamos esta matriz para anticipar problemas:

Riesgo Detectado	Tipo	Impacto	Probabilidad	Estrategia de Respuesta
Incumplimiento GDPR/LOPD	Legal	**Crítico**	Media	**Mitigar:** Implementar anonimización de datos en la capa de ingesta. Los datos personales (DNI, Nombre) se separarán de los datos de consumo mediante *hashing*.
Cuello de botella en Ingesta	Técnico	Alto	Alta	**Transferir/Mitigar:** Uso de arquitectura distribuida (Kafka). Realizar pruebas de estrés (*Stress Testing*) simulando el doble de carga prevista antes del lanzamiento.
Mala Calidad del Dato	Datos	Alto	Alta	**Mitigar:** Implementar reglas de validación en la entrada (ej. una casa no puede consumir 0kw en un mes ni 1GW en una hora). Los datos erróneos irán a una "Dead Letter Queue" para revisión manual.
Sobrecoste Cloud	Financiero	Medio	Media	**Evitar:** Configurar alertas de presupuesto en AWS/Azure y usar instancias reservadas para las cargas base, dejando el pago por uso solo para picos.
Falta de talento especializado	RRHH	Alto	Media	**Aceptar/Mitigar:** Contratación de consultoría externa para el arranque y plan de formación interna en Scala y Spark para los desarrolladores actuales.

0 5
A R Q U I T E C T U R A T É C N I C A P R O P U E S T A

Entramos en el centro del trabajo. Proponemos una Arquitectura Lambda. Es el estándar para manejar datos históricos y datos en vivo a la vez. Todo reside en la nube para permitir el crecimiento.

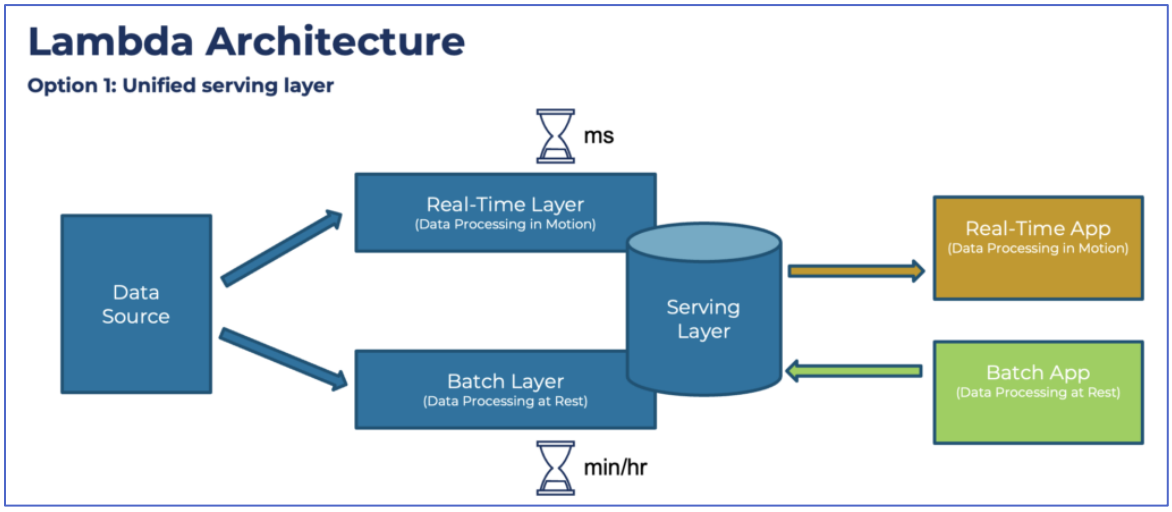
La estructura tiene 4 capas:

A. Capa de Ingesta (La puerta de entrada)

Tecnología: Apache Kafka.
Por qué: Funciona como un buffer. Separa los contadores del sistema central. Kafka guarda los mensajes si el sistema se satura. No perdemos ninguna lectura.

B. Capa de Almacenamiento (Data Lake)

Tecnología: HDFS o Amazon S3.
Estrategia: Guardamos todo.
Zona Raw: Contiene el dato original "sucio" (JSON/CSV).
Zona Refined: Contiene el dato limpio en formato Parquet o Avro. Estos formatos leen mucho más rápido.

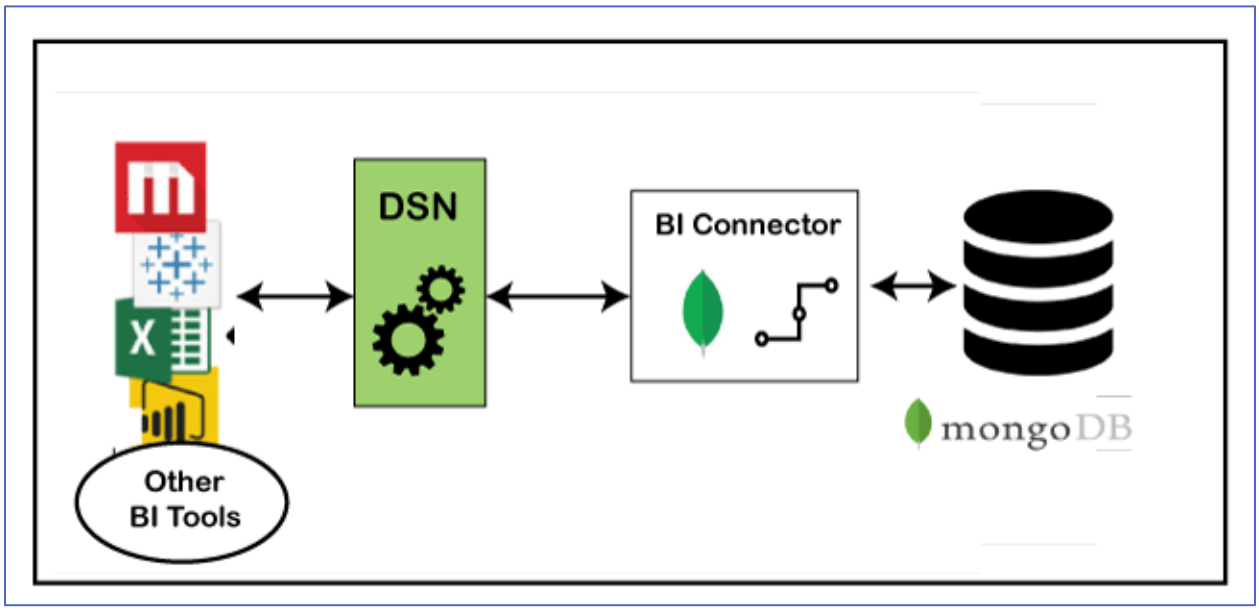


C. Capa de Procesamiento (El cerebro)

Tecnología: Apache Spark (programado en Scala).
Qué hace:
Ejecuta procesos ETL nocturnos para limpiar y ordenar.
Realiza análisis avanzado. Usaremos Spark MLlib para algoritmos de agrupación (K-Means). La máquina clasifica clientes por comportamiento (ejemplo: "Los nocturnos").

D. Capa de Visualización (Serving Layer)

Tecnología: MongoDB para la aplicación y Power BI para los directivos.
Por qué: MongoDB muestra el consumo al usuario en milisegundos. Power BI conecta al Data Lake y crea gráficos de tendencias anuales.



0 6
F A S E S D E E J E C U C I Ó N

Seguimos el ciclo de vida del dato para mantener el orden:

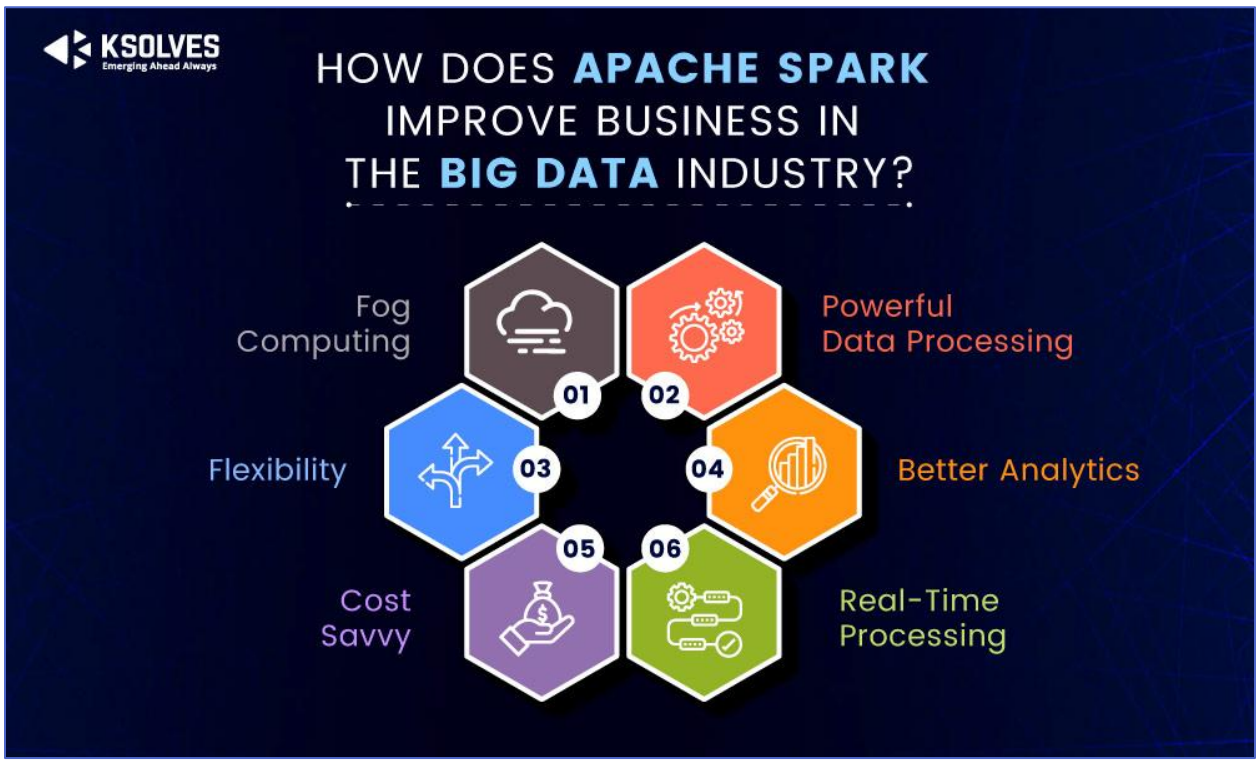
1. **Descubrimiento (Sprint 0):** Definimos "cliente rentable" junto con el cliente.

2. **Ingeniería de Datos:** Montamos el Clúster. Conectamos los primeros 1.000 contadores de prueba.

3. **Ciencia de Datos (Modelado):** Entrenamos modelos con datos antiguos anónimos. Comprobamos el funcionamiento del algoritmo K-Means.

4. **Despliegue:** Pasamos los modelos a producción sobre Spark.

5. **Monitorización:** Medimos el acierto de las recomendaciones y hacemos ajustes.



0 7
C O N C L U S I O N E S

El diseño de esta plataforma eléctrica me enseñó mucho. La modernización digital es compleja por dentro. No basta con instalar Hadoop. La forma de trabajar debe cambiar.

La escalabilidad es el punto fuerte de nuestra propuesta. El uso de **Kafka** y **Spark** en la nube hace al sistema fuerte. Las metodologías ágiles permiten reacciones rápidas ante cambios de precio o mercado. También insistimos mucho en la seguridad (GDPR) debido a las exigencias de este sector. En el sector, la confianza del cliente lo es todo. Unir todas las piezas ha sido un reto técnico. Creemos que es la arquitectura más sólida para el futuro de la compañía.

0 8
R E F E R E N C I A S

https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf

https://www.atlassian.com/es/agile/kanban/kanban-vs-scrum

https://spark.apache.org/docs/latest/

https://www.confluent.io/what-is-apache-kafka/

https://aws.amazon.com/es/what-is/data-lake/

https://www.youtube.com/watch?v=1vbXmCrkT3Y

https://www.youtube.com/watch?v=HhC75IonpOU